Appeal Nos. 14-1542, -1543

United States Court of Appeals
for the
Federal Circuit

MICROSOFT CORPORATION,

Appellant,

– v. –

PROXYCONN, INC.,

Cross-Appellant.

APPEAL FROM THE UNITED STATES PATENT AND TRADEMARK OFFICE, PATENT TRIAL AND APPEAL BOARD, IN CASE NOS. IPR2012-00026, IPR2013-00109

OPENING BRIEF OF APPELLANT MICROSOFT CORPORATION

JOHN D. VANDENBERG
STEPHEN J. JONCUS
CARLA TODENHAGEN
KLARQUIST SPARKMAN, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 595-5300
john.vandenberg@klarquist.com

Attorneys for Appellant Microsoft Corporation

August 8, 2014

FORM 9. Certificate of Interest

UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

MICROSOFT CORPORATION v. PROXYCONN, INC.

No. 14-1542

CERTIFICATE OF INTEREST

Counsel for the (petitioner) (appellant) (respondent) (appellee) (amicus) (name of party) Microsoft Corporation certifies the following (use "None" if applicable; use extra sheets if necessary):

1. The full name of every party or amicus represented by me is:

Microsoft Corporation

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

None.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are:

None.

4. ☑ The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court are:

KLARQUIST SPARKMAN, LLP: John D. Vandenberg, Salumeh R. Loesch, Stephen J. Joncus, and Carla Todenhagen.

June 25, 2014
Date

/s/John D. Vandenberg
Signature of counsel

John D. Vandenberg
Printed name of counsel

Please Note: All questions must be answered
cc: Matthew L. Cutler

TABLE OF CONTENTS

Page

Case: 14-1542 Document: 12 Page: 5 Filed: 08/08/2014

TABLE OF AUTHORITIES

Page(s)

Cases

Case: 14-1542 Document: 12 Page: 6 Filed: 08/08/2014

Other Authorities

Regulations

Rules

Statutes

STATEMENT OF RELATED CASES

Pursuant to Fed. Cir. R. 47.5, counsel for Appellant, Microsoft Corporation, states as follows:

(a) No other appeal in or from the same proceeding was previously before this or any other appellate court whether under the same or a similar title; and

(b) There are no related cases pending before this Court or any other court.

JURISDICTIONAL STATEMENT

This is an appeal pursuant to 35 U.S.C. § 141(c) from the Final Written Decision (Paper 73) of the Patent Trial and Appeal Board of the United States Patent and Trademark Office entered on February 19, 2014 in *Inter Partes* Review Nos. IPR2012-00026 and IPR2013-00109.

Microsoft Corporation timely filed a notice of appeal on April 22, 2014. This Court therefore has jurisdiction over this appeal under 35 U.S.C. § 141 and 28 U.S.C. § 1295(a)(4)(A).

STATEMENT OF THE ISSUES

This appeal presents a narrow legal issue specific to this case: what is the "broadest reasonable interpretation" (BRI) of the claim term "searching" in claim 22 of Proxyconn's patent? But it also presents a broader issue of

law of importance to Patent Office patent-review proceedings generally: how should the Board use a patent specification and dictionaries in reaching the "broadest reasonable interpretation"? Here, finding in the patent no "explicit" definition of the claim term "searching," the Board proceeded to rely exclusively on dictionaries, ignoring the patent specification. That was legal error, and the error led the Board to an erroneous claim interpretation that is so narrow that it excludes all embodiments in the patent. Microsoft asks this Court to reject the Board's erroneous construction and erroneous analytical approach, vacate the Board's determination that claim 24 is patentable even though all the other challenged claims are not, and remand for further proceedings under a correct BRI of this claim language.

STATEMENT OF THE CASE

A. The Problem Of Verifying That A Locally Cached Document Is Up-To-Date, And The Patent's Solution

The challenged patent, U.S. Patent No. 6,757,717 ("the '717 patent") (A00070-88), addresses a need faced by anyone who wants to use a locally stored document (e.g., a draft legal brief on an attorney's laptop computer) that may or may not be current with the master version stored elsewhere (e.g., on a law firm's main office server computer). The need is to verify that the locally saved ("cached") copy of the document is not out-of-date. In other words, the system must determine whether the locally stored copy is

synchronized with the corresponding remotely stored version or, instead, obsolete. To be safe, one could repeatedly retrieve the master copy from the main office server each time the document is needed locally, but that would waste time and network bandwidth whenever the local copy was already up-to-date.

Figure 1 of the patent shows the environment for this situation, illustrating that the local client Receiver's cache memory is separated by a network from a remote server Sender's copy:

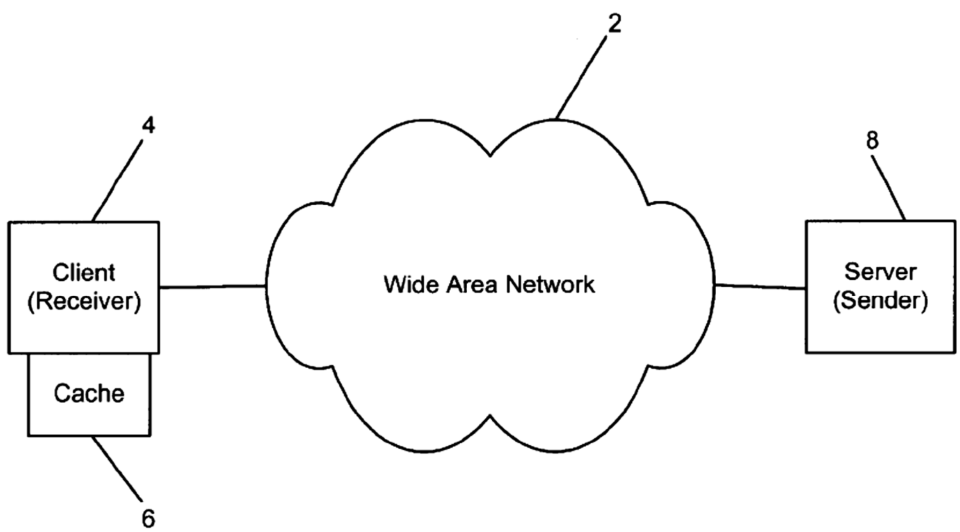


FIG. 1 (PRIOR ART)

A00071. In our example, the document stored locally may be a draft of this Brief on an attorney's laptop computer (the client or "Receiver") and the master version may be stored on a law firm's file server (the "Sender").

Referring to Fig. 1, the patent first explains that it was known to cache data locally for possible reuse:

> Many known applications and protocols provide means for caching and verifying of data transmitted via a network 2 (FIG. 1, prior art). Thus, a client (receiver) 4 caches data received from network 2 in cache 6. Then, when data from a remote server (sender) 8 is requested, it first searches its local cache. If the requested data is available in the cache and is verified to be valid, the client uses it, and transmission over the network is not required.

A00080 (1:18-25).

The patent then explains some challenges posed by the need to verify that the locally stored (cached) copy of the data object is up-to-date:

> The problem associated with this technique is that the data entity can be changed before the expiration time, and the receiver would use an obsolete version of the data without even knowing it. Also, when the time has expired, the data will be resent, even if it is up to date.

A00080 (1:35-39).

The patent's proposed solution to this obsolescence-detection problem relies on calculating a relatively short digital "fingerprint" (called a "digest") of the content of a document (called a "data object"). This fingerprint or digest uniquely (within accepted probabilities) represents that particular data object's contents. Thus, if two digests are identical, then the contents of their respective data objects are likewise identical. Conversely, if two digests are different then their respective data objects are different too.

Digests are much smaller than their corresponding data objects. Accordingly, communicating digests instead of data objects over a network reduces use of bandwidth.

In one use scenario, the local Receiver indicates a need to verify that a particular data object is up-to-date. The remote Sender (e.g., main office server) then transmits a digest of the master version of that same data object over the network. The local Receiver then checks to see if it has the same digest value—and thus the same unmodified data object—in its cache. It does this search for a matching digest value by comparing the digest received from the Sender to its own digest of its local copy of the same data object. If those two digests match, the Receiver can be confident that its copy of the data object of interest is up-to-date, and it can use it. But when the two digests do not match, the Receiver so notifies the Sender, which then sends the up-to-date copy of the data object over the network to the Receiver. This way, updates of data objects are sent only when they are needed.

Proxyconn's expert, Dr. Konchitsky, summarized this operation in his direct testimony, as follows: "in the system claimed in claims 1 and 3 and with the method claimed in claims 22-24, a request is sent and digital digests

are used to determine whether the receiver computer has the file, or whether it needs to be provided by the sender/computer." A00502-3 (¶ 17).

B. Claim 24 And Its "Searching" Limitation

The claim language disputed in this appeal appears in the italicized portion of claim 22, from which the claim at issue (claim 24) depends:

> 22. A method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of:
>
> receiving a message containing a digital digest from said network;
>
> *searching for data with the same digital digest in said network cache memory*,
>
> if data having the same digital digest as the digital digest received is not uncovered, forming a negative indication signal and transmitting it back through said network; and
>
> creating a digital digest for data received from said network cache memory.

A00085 (12:30-45). Claim 24 adds the additional requirement that "a plurality of digital digests for different data objects is received in the same message and an indication signal is generated separately for each of said data objects," but that additional limitation is not at issue in this appeal.

Some of these steps are depicted in Fig. 5 of the patent:

Case: 14-1542 Document: 12 Page: 13 Filed: 08/08/2014

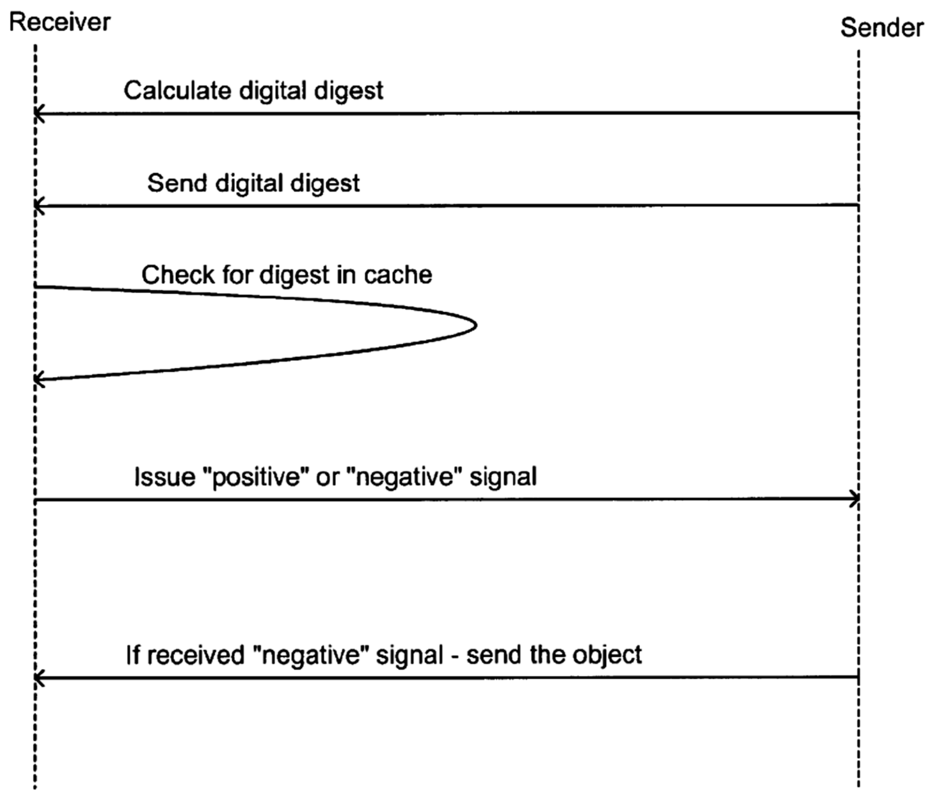


FIG. 5

A00073. The "receiving" step in claim 22 relates to "Send digital digest" in Fig. 5; the "searching" step in claim 22 corresponds to "Check for digest in cache;" and "forming a negative indication signal" in claim 22 corresponds to "Issue … 'negative' signal" in Fig. 5.

C. The Patent's Equation Of "Searching" And "Checking For"

As described above, part of the patent's algorithm is to check whether a matching digest of the data object appears in the local Receiver's cache memory. The patent calls this operation both "check for" and "search." Thus, the step that Fig. 5 calls "check for digest in cache," the written

description calls "searches its network cache memory 48 for data with the same digest" A00083 (7:56-57).

As noted, the patent's algorithm performs this *search* by comparing the Sender's digest for the data object with the Receiver's digest for the *same* data object to determine whether they match. But before the Sender's digest can be compared to the corresponding Receiver's digest for that data object, the Receiver's particular digest must be located. The patent does not describe this preliminary step of locating and retrieving the Receiver's digest for the data object of interest; it simply assumes locating the digest for the corresponding data object. The closest the patent comes to discussing this preliminary step of locating the local digest to compare to the received digest is when it states that the Sender may tell the Receiver the possible location for the Receiver's digest. A00083 (8:53-56). The only part of the "search" operation actually described in the patent is the single digest-to-digest comparison operation.

D. The Patent Does Not Disclose Performing Multiple Digest Comparison Operations

As explained above, the patent's algorithm requires only a single digest-to-digest comparison operation to determine whether or not its local copy of the desired data object is up-to-date. The patent does *not* describe searching for matches beyond the single digest of the single data object

corresponding to the received digest. And it does not describe—in this context—comparing a received digest to more than one locally stored digest.[1]

Indeed, given the obsolescence-detection problem addressed by the patent, trying to match the received digest to more than one digest on the Receiver would be a waste of time and resources. That is because there is only one candidate for an exact match: the Receiver's digest of the *same* data object from which the received digest was calculated. And the patent does not describe finding that one candidate through a digest-comparison operation.

A hypothetical illustrates why it makes sense that the patent's algorithm requires only a single digest-to-digest comparison operation in view of the obsolescence-detection problem addressed. Assume that the digital digest received by the Receiver over the network is a digest of the master copy of a draft of this Brief stored on a main office's server computer. Comparing that digest of the master copy to the locally cached digest *of this Brief* on the attorney's laptop makes sense, to verify that the local copy is up-to-date. But comparing it to the locally cached digest of a

[1] The patent does describe a different scenario (Fig. 8) in which differences between versions of the same object are determined and downloaded, but claim 24 is not directed to that alternative scenario.

lunch menu, an Excel spreadsheet, or any other document on the attorney's laptop computer does not make sense. Given this patent's obsolescence-detection goal, the "search" for a matching data object ends with a comparison between a *single* pair of digests, namely the Sender's and Receiver's respective digests of the same single data object that the user wants to verify as being up-to-date.

Of course, in many other search contexts, it does make sense to check for matches in multiple locations. For example, a typical Web search using a search engine will look for and return scores of matches from scores of Web sites. The keywords entered by a user in her search request are not, typically, unique to a single page on the Web. But that is not the context here. This patent does not disclose a scenario in which two objects with different names have the same data content, and it does not disclose searching for such different-name, same-content objects. In this context, the user does not want multiple matches. Instead, the user simply wants to know if her local copy of the desired file is up-to-date. As Proxyconn's expert explained, "digital digests are used to determine whether the receiver computer has the file, or whether it needs to be provided by the sender/computer." A00502-3 (¶ 17).

In sum, the patent does not disclose any "search" step that performs more than one digest-to-digest comparison operation.

E. Proxyconn's Expert's Confirmation Of The Kind Of "Searching" Contemplated By This Patent

The cross-examination testimony of Proxyconn's expert, Dr. Konchitsky, confirms the nature of the "searching" contemplated by this patent. Dr. Konchitsky distinguished the type of "search" in this patent from the different type of "search" performed by the Google Web search engine. A00669 (48:5-9). In this patent, the "search" determines whether a single local copy of a digest does or does not match a single received digest for the same, unique data object. A00669-10 (48:11-49:7). Dr. Konchitsky thus agreed that a search in this patent does *not* require looking at multiple digest values:

> Q. Does the '717 patent require that as part of the search for data with the same digest that the receiver look at multiple digest values that it has stored?
>
> MR. WHEELOCK: Objection. Form. Scope.
>
> BY MR. VANDENBERG: Q. Does it require that?
>
> A. As of now, I see that it could look at one or many or all of them, but *it's not a requirement*.

A00671 (54:4-12) (emphasis added).

On the contrary, he testified, the patent describes using a comparison between two digests to check for a matching digest. A00674 (67:7-12). More

specifically, if the receiver "has the digest, it just checks the digest and see [sic] if this particular value matches to the other value, so that is how the positive or negative signal [indicating match or no match] will be issued." A00672-3 (62:24-63:8); A00673 (63:14-23).

F. Microsoft's Proposed Interpretation Of "Search"

Proxyconn did not expressly propose an interpretation of the claim term "searching." It implied, however, that "searching" requires more than one digest-to-digest comparison. A00603; A00610-1. Microsoft disagreed. Relying primarily on the specification and the admissions of Dr. Konchitsky, Microsoft contended that the broadest reasonable interpretation of "search" was "[c]heck for," with the corollary that "[a] 'search' does not require checking multiple digest values for a match." A00636-7.

G. The Prior Art's Similar Use Of A Single Digest-To-Digest Comparison To Check For Matching Data

Both prior art references asserted against claim 24 disclosed the same digital-digest algorithm described in the '717 patent, including its single digest-to-digest comparison operation to check for matching cached data.

U.S. Patent No. 5,742,820 to Perlman *et al.*, issued in 1998 and entitled "Mechanism for Efficiently Synchronizing Information Over a Network," called its digest a "database identifier" and described this same digital-digest algorithm as follows:

> The invention comprises a mechanism for efficiently synchronizing the contents of databases stored on nodes of a computer network to ensure that those contents are consistent. Generally, the mechanism comprises a database identifier generated by a node of the computer network and distributed to other receiving nodes coupled to the network. The database identifier is uniquely representative of the contents of the distributing node's database and the *receiving nodes compare this unique identifier with their own generated database identifiers to determine if the identifiers, and thus their databases, are consistent and synchronized*.

A00143 (3:61-4:4) (emphasis added); *see also* A00145-6 (8:52-9:2); A00146 (claims 1, 4-6, 10).

In Perlman, each receiver stored in its cache memory a database of data objects called "LSP data items." A00144 (6:25-56), A00138 (Fig. 3). Before Perlman, such systems synchronized these LSP databases by having a sender send to each receiver "CSNP fragments" identifying portions of the LSP data items in the sender's database. A00144 (6:46-56). Each receiver compared those received CSNP fragments to its own LSP database and requested updates as needed. *Id.* (6:57-64). But that technique for detecting out-of-date data was inefficient: "in order to characterize an entire LSP database, the CSNP may be very large" (*id.* (6:65-66)), and transmission of the CSNP fragments "consumes significant bandwidth." A00145 (7:7-11). Perlman's improvement had the sender instead send short digests (database identifiers) of each CSNP fragment, rather than the longer fragment itself.

These digests were "identifiers for particular fragments of the database" (*id.* (8:36-38)), and they were sent in messages containing multiple identifiers (digests), each corresponding to a particular portion of the LSP database (A00141 (Fig. 7)).

For each individual digest received over the network, a Perlman receiver found its own digest for the corresponding data object and compared it to the received digest. A match from that one-to-one digest comparison operation indicated that the corresponding portion of the receiver's LSP database was up-to-date. A mismatch indicated that this portion of the database was not up-to-date. A00145 (8:7-42), A00145-6 (8:52-9:2). Thus, using only *one* digest-to-digest comparison operation, Perlman was able to identify a particular data object (a particular portion of the LSP database) with the same digital digest out of a set of potentially many data objects (all other portions of the LSP database) stored in the network cache memory.[2]

---

[2] Although Perlman's illustrative embodiment was a network-router embodiment in which the data objects described aspects of the network, his design was not limited to that embodiment. Perlman expressly disclosed (A00135, Abstract; A00142 (1:1-9); A00143 (3:60-4:4); A00145 (7:12-22); A00145-6 (8:52-9:2)) and claimed (A00146 (claims 1 and 4-6)) that its mechanism applies to any type of distributed system and network node requiring such database synchronization.

U.S. Patent No. 5,835,943 to Yohe *et al.*, issued in 1998, and entitled "Apparatus and Method for Increased Data Access in a Network File Oriented Caching System," disclosed the same algorithm and the same ability to identify a particular data object with the same digital digest from a set of potentially many data objects stored in the receiver's network cache memory, using only a single digest-to-digest comparison. A00173 (2:41-61); *id.* (2:58-61); A00175 (5:1-3); A00176 (8:9-11); *id.* (8:13-21); A00179 (13:34-35); A00179-80 (14:40-15:3); A00160 (Abstract); A00171 (Fig. 15); A00179-80 (claims 1, 8). As the Board recognized in this case, Yohe disclosed "a series of one-to-one comparisons of received digests to locally generated digests," one comparison for each data object in question. A00039; *see also* 1st Declaration of Microsoft Expert Dr. Long, A00190 (6-13); Deposition Transcript of Dr. Long, A00916 (183:3-21).

H. Microsoft's *Inter Partes* Review Petitions And The PTAB's Ruling

This case began in 2012 when Microsoft filed a petition for *inter partes* review challenging the patentability of various claims of the '717 patent that Proxyconn had asserted against it in district court litigation. A00093. The Board agreed to consider challenges to certain claims including claims 22-24. A00353-4. Microsoft later filed a second petition

challenging additional claims newly asserted by Proxyconn, and the Board joined the two proceedings. A00359; A00428.

In its final decision, the Board determined that all the reviewed claims except claim 24 were unpatentable as anticipated and that three claims were also obvious. A00064. The Board accordingly issued an order canceling all the reviewed claims except claim 24. *Id*. Because the Board's rulings regarding claims 1, 3, 6, 7, 9-12, 14, 22, and 23 are the subject of Proxyconn's cross-appeal, Microsoft will defer its discussion of those rulings until its response on cross-appeal.

Microsoft contended that dependent claim 24 was anticipated by Perlman or at least obvious in light of Perlman and Yohe. The Board disagreed, finding that neither Perlman nor Yohe disclosed the "searching" limitation of underlying claim 22. A00052-53.[3] Those determinations were based on the Board's interpretation of the "searching" limitation.

The Board interpreted "searching" to "require[ ] an ability to identify a particular data object with the same digital digest from a set of potentially many data objects stored in the network cache memory." A00019-20.

---

[3] The Board found that another reference, Santos, fully anticipated claim 22, including the "searching" limitation. A00043. The Board did not consider whether claim 24 was unpatentable in view of a combination of Santos with Perlman and/or Yohe because Microsoft's petition did not raise those particular combinations.

Although the exact scope of this interpretation is unclear, the Board was clear that its interpretation *excluded* comparing the received digest to only the *single* corresponding local digest for the same data object:

> We conclude that Microsoft has failed to establish by a preponderance of evidence that Perlman meets the "searching" step recited in claim 22. Every portion of Perlman upon which Microsoft relies involves comparing one received digest to a corresponding locally calculated digest. These one-to-one comparisons cannot identify a particular data object from among a set of data objects as required by the "searching" step. Rather, the comparisons reveal whether a locally stored data object is synchronized with the corresponding remotely stored data object.

A00036 (*see also* A00039-40, distinguishing Yohe prior art on same ground).

The Board acknowledged "Microsoft contends that the '717 patent equates 'search' with 'check for' and that the Specification never describes any 'search' method other than 'comparing two digest values for a match.' MS Reply 4." A00019. But the Board did not attempt to reconcile its narrow interpretation of "searching" with the different manner in which the patent's preferred embodiment operates. Nor did it acknowledge the testimony of Proxyconn's expert, Dr. Konchitsky, about how the claimed invention worked: if the receiver "has the digest, it just checks the digest and see [sic] if this particular value matches to the other value, so that is how the positive

Case: 14-1542 Document: 12 Page: 24 Filed: 08/08/2014

or negative signal will be issued." A00672-3 (62:24-63:8); A00673 (63:14-23).

Instead, the panel dismissed the specification because it "never expressly defines 'search'" and relied exclusively on dictionary definitions that had not been cited in the case:

> The Specification never expressly defines "search." Nonetheless, the plain meaning of "search" is: "to look into or over carefully or thoroughly in an effort to find or discover something." MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 1053 (10th ed. 1999). Two dictionaries in the relevant field of computing technology define "search" as it would be understood by a skilled artisan as follows:
>
> 1. "To scan one or more data elements of a set in order to find elements that have a certain property," IBM DICTIONARY OF COMPUTING 600 (10th ed. 1993); and
> 2. "(information processing). To examine a set of items for those that have a desired property," IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONICS TERMS 808 (3d ed. 1984).
>
> These dictionary definitions reflect that a skilled artisan would have understood "search" to involve analyzing a set of items to identify one particular item from among a set of items. A "set" refers to "a number of things of the same kind that belong or are used together," MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 1071 (10th ed. 1999), or "[a] finite or infinite number of objects of any kind, of entities, or of concepts that have a given property or properties in common," IBM DICTIONARY OF COMPUTING 618 (10th ed. 1993). While a set can contain one item, a "search" for a desired member of a "set" requires a capability to examine more than one item to identify a particular item within that set. Therefore, we conclude that

> "searching for data with the same digital digest in said network cache memory" requires an ability to identify a particular data object with the same digital digest from a set of potentially many data objects stored in the network cache memory.

A00020-21.

SUMMARY OF ARGUMENT

The Board committed several errors of law in determining a "broadest reasonable interpretation" (BRI) of the limitation of claim 22 requiring "searching for data with the same digital digest in said network cache memory." The Board's erroneous claim interpretation caused it to reject Microsoft's unpatentability arguments against dependent claim 24. This Court should therefore vacate the Board's judgment on claim 24 and remand for reconsideration of the patentability of claim 24 under the proper, broader construction.

First, the Board chose a narrower interpretation over the reasonable and broader interpretation proffered by Microsoft. This is contrary to the BRI rule that applies to *inter partes* reviews.

Second, the Board determined the "plain meaning" of the claim term in a non-contextual way by considering only dictionaries and not the patent's specification. That was an error of law because "plain meaning" is a short-

hand for the meaning the term would have to a person of skill in the art *in light of and after reading the patent*.

Third, the Board erroneously limited the role of the specification in its BRI analysis to being no more than a possible source of *express* definitions. This demotion of the specification was legal error.

Fourth, the Board erroneously read into the claim a limitation—performing *multiple* digest-to-digest comparison operations as part of a search—that is not even mentioned in the specification. To narrow the meaning of the claim by reading in a non-disclosed limitation is legal error.

Fifth, the Board's interpretation is too unclear to be deemed reasonable. Specifically, the Board's interpretation that "searching" "requires an ability to identify a particular data object with the same digital digest from a set of potentially many data objects stored in the network cache memory"—improperly conflates a method step—which is an action—with a capability, which is a function of a device. This renders the interpretation unclear.

These errors of law were not harmless. On the contrary, the Board expressly relied on its narrower and legally erroneous interpretation of this claim step to reject each of Microsoft's urged grounds of unpatentability of claim 24.

The Court should reject the Board's non-contextual, unsupported, narrow, and unclear interpretation and instead give the claim term "searching" (and associated method step in claim 22) the broader and more reasonable interpretation that Microsoft proffered. The Court should also vacate the Board's resulting ruling that Microsoft failed to show unpatentability of claim 24 and remand for the Board to determine patentability under the correct claim interpretation.

ARGUMENT

A. This Is A *De Novo* Review Of The Board's Broadest Reasonable Interpretation

The Court reviews the Board's BRI and other legal conclusions *de novo*. *Tempo Lighting, Inc. v. Tivoli, LLC*, 742 F.3d 973, 976-77 (Fed. Cir. 2014). Applying that standard, the Court should reverse the Board's interpretation of the "searching" step of claim 22 and remand for the Board to reconsider the patentability of claim 24.

B. Microsoft's Interpretation Of The "Searching" Step Is The Broadest Reasonable Interpretation

Under the Board's regulations governing *inter partes* reviews, "[a] claim in an unexpired patent shall be given its broadest reasonable construction in light of the specification of the patent in which it appears." 37 C.F.R. § 42.100(b). The broadest reasonable interpretation of "search" in

this patent is that proffered by Microsoft: "check for," with no requirement of checking multiple digest values for a match.

Microsoft's interpretation is consistent with the claims and specification, consistent with the testimony of Proxyconn's expert, Dr. Konchitsky, and even consistent with the second definition in the IBM dictionary cited by the Board. The Board did not declare Microsoft's proposal unreasonable—and could not have done so.

The analytical starting point is the claim. Claim 22 recites a method performed by a Receiver. The Receiver's first step is receiving "a digital digest" from a network. In this patent, a "digital digest" is calculated from a single, particular data object, and there "is a low probability that two different data or objects have the same digital digest." A00341; A00080 (2:9-13). This is why such a digest is also called a "fingerprint." Because each data object has a unique digest, there is only one plausible candidate for a matching digest on the Receiver, namely the Receiver's digest of *that same data object* from which the received digest had been calculated. The claim's second step is the step at issue on this appeal: "searching for data with the same digital digest in [the receiver's] network cache memory." Again, there is only one plausible candidate object with the same digest as the received digest. Therefore, only that digest needs to be checked to complete this

search operation. The Receiver's third step is a conditional one: *if* data with the same digest is *not* found, forming and transmitting through the network a "negative indication signal."

The claim language does not expressly limit how the searching step is performed. It does not, for example, require that the received digital digest be compared to multiple digests stored at the Receiver, including digests calculated for different data objects than the one data object being verified.

Each claim, of course, is part of the specification. 35 U.S.C. § 112, ¶ 2. The specification's Fig. 5 refers to the claim's second, searching step as "check for digest in cache." That implicit definition in the specification provides strong support for Microsoft's position that "search" means "check for."

The function of the searching step also supports Microsoft's proposed interpretation. *See Medrad, Inc. v. MRI Devices Corp.,* 401 F.3d 1313, 1319 (Fed. Cir. 2005) (It is "entirely proper to consider the functions of an invention in seeking to determine the meaning of particular claim language."). The function here is to determine whether the data object copy in the Receiver's cache is the same as or different from the copy at the remote Sender site. This is determined by comparing fingerprints (digests) of that pair of copies. The specification discloses no reason relevant to claim 24

to compare the data object's digest to a digest for a *different* data object. In this context, only one data item in the Receiver is a plausible candidate for a match. Therefore, all that is needed to satisfy the function of this searching operation is to retrieve that candidate digest and perform a single comparison with that digest. This function of the claim element therefore supports the remainder of Microsoft's proposed BRI: a "search" does *not* require checking multiple digest values for a match.

As explained above (at Statement Of The Case (E)), Proxyconn's expert, Dr. Konchitsky, agreed with Microsoft's reading of the specification. As he testified, if the Receiver "has the digest, it just checks the digest and see [sic] if this particular value matches to the other value, so that is how the positive or negative signal will be issued." A00672-3 (62:24-63:8); A00673 (63:14-23). In other words, if the Receiver "has the digest" associated with the data object in question, then it just compares that digest to the received digest and based on the results of that single comparison operation, determines whether its local copy of the data object does or does not need to be updated. This too supports Microsoft's proposed BRI. And no evidence of record conflicts with Microsoft's proposed BRI.

Case: 14-1542 Document: 12 Page: 31 Filed: 08/08/2014

C. The Board Committed Legal Error By Dismissing A Broader Reasonable Interpretation In Favor Of A Narrower Interpretation

The Board did not and could not find Microsoft's interpretation unreasonable. Moreover, the Board's interpretation is unquestionably narrower than Microsoft's. The Board limited "searching" to a procedure that performs multiple digest-to-digest comparison operations, at least when the first comparison does not find a match, while Microsoft's interpretation has no such limitation. Because Microsoft's interpretation is reasonable, it is the broadest reasonable interpretation on this record and must be adopted under the Board's own rules.

By picking a narrower interpretation over a reasonable and broader alternative, the Board committed legal error. *Tempo Lighting*, 742 F.3d at 977-78 (rejecting the Examiner's narrow interpretation based on dictionary definitions; "the examiner erred by resorting to extrinsic evidence that was inconsistent with the more reliable intrinsic evidence."); *Leo Pharm. Products, Ltd. v. Rea*, 726 F.3d 1346, 1352-1353 (Fed. Cir. 2013) ("The Board erred by narrowing the definition of 'storage stable' to something far short of its broadest reasonable meaning.").

D. The Board Committed Legal Error By Ignoring The Specification When Determining "Plain Meaning"

The Board also committed legal error by relying entirely on dictionary definitions and ignoring the patent's specification, when it determined the "plain meaning" of the claim language. The Board's decision here aptly illustrates why relying on dictionary definitions divorced from the context of the patent is an unreliable approach to claim interpretation.

"Properly viewed, the 'ordinary meaning' of a claim term is its meaning to the ordinary artisan *after* reading the entire patent." *Phillips v. AWH Corp.,* 415 F.3d 1303, 1321 (Fed. Cir. 2005) (en banc) (emphasis added); *see Lexion Med., LLC v. Northgate Tech., Inc.*, 641 F.3d 1352, 1356 (Fed. Cir. 2011) ("The customary meaning of a claim term is not determined in a vacuum and should be harmonized, to the extent possible, with the intrinsic record, as understood within the technological field of the invention."); *Medrad,* 401 F.3d at 1319 ("We cannot look at the ordinary meaning of the term ... in a vacuum. Rather, we must look at the ordinary meaning in the context of the written description and the prosecution history."); *Novartis Pharm. Corp. v. Abbott Labs.*, 375 F.3d 1328, 1335 (Fed. Cir. 2004) (extrinsic evidence "may not be used to vary, contradict, expand, or limit the claim language from how it is defined, even by implication, in the specification or file history"); *cf. Towne v. Eisner*, 245

U.S. 418, 425 (1918) ("A word is not a crystal, transparent and unchanged, it is the skin of a living thought and may vary greatly in color and content according to the circumstances and the time in which it is used.").

All the reasons why "plain meaning" is determined *in context* in the courts apply equally to the Patent Office. Indeed, the Patent Office recognizes this in its guidance to Examiners. *See* MPEP § 2111.01 (I) ("The ordinary and customary meaning of a term may be evidenced by a variety of sources, including the words of the claims themselves, the specification, drawings, and prior art. However, the best source for determining the meaning of a claim term is the specification—the greatest clarity is obtained when the specification serves as a glossary for the claim terms.").

Reading claim language in context is especially important where that language has multiple meanings or connotations. "Search" is such a term. For example, both technical dictionary definitions of "search" cited by the Board concern finding *multiple* elements having a certain property:

> 1. "To scan one or more data elements of a set in order *to find elements* that have a certain property," IBM DICTIONARY OF COMPUTING 600 (10th ed. 1993); and
>
> 2. "(information processing). To examine a set of items *for those* that have a desired property," IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONICS TERMS 808 (3d ed. 1984).

A00020 (emphases added).

These definitions thus concern the common search situation where one seeks multiple matches, as in conducting a search on the Web. But that is *not* the scenario addressed in this patent. In this patent, the goal of the "search" is to check whether a *single* unique item (e.g., the draft of this Brief) is unchanged. "Search" in this patent indisputably is *not* a search for multiple elements. The dictionary definitions selected by the Board address a different context and thus are inapt here.

The Board's error becomes even more apparent when one reads the alternative dictionary definitions that the Board did *not* select (or, indeed, mention). Specifically, from the *IBM Dictionary of Computing* it found and cited, the Board chose, without explanation, the arguably narrowest of three alternative definitions:

> search (1) A function or mode that enables the user to locate occurrences of such things as particular character strings, embedded commands, or boldface letters in a document. Synonymous with find. (T) (2) The process of looking for a particular item. See also browse, scan. (3) *To scan one or more data elements of a set in order to find elements that have a certain property*. (4) See binary search, chaining search, dichotomizing search, Fibonacci search.

A01149 (emphasis added).

At least the second definition ("looking for a particular item")—not mentioned by the Board—is broader. It does not require searching for multiple elements and certainly does not require performing multiple digest

Case: 14-1542 Document: 12 Page: 35 Filed: 08/08/2014

comparisons. In the context of this patent, the "particular item" is the current version of the desired file, and the process of "looking for" is accomplished by comparing the digest for that current version with the Receiver's digest for that file. This broader, second definition therefore is more apt than the narrower, third definition selected by the Board, which relates to a very different search context.

Exacerbating this error, the Board went outside the record and *sua sponte* found and selected these particular definitions with no notice to the parties. Neither party suggested these dictionary definitions or any equivalent dictionary definition. Instead, Microsoft cited the specification and the testimony of Proxyconn's expert witness about the meaning of "search" in this patent specification. To construe a patent claim based entirely on non-record evidence—to the exclusion of the specification and expert testimony actually cited by the parties—does not conform with the governing rules of evidence, or basic fairness. *Cf.* Fed. R. Evid. 201; 37 C.F.R. § 42.62; *Phillips*, 415 F.3d at 1322 ("A claim should not rise or fall based upon the preferences of a particular dictionary editor, or the court's independent decision, uninformed by the specification, to rely on one dictionary rather than another.").

Case: 14-1542 Document: 12 Page: 36 Filed: 08/08/2014

E. The Board Committed Legal Error By Ignoring The Specification's Teachings When Determining The "Broadest Reasonable Interpretation"

The Board did not merely fail to consider the specification in arriving at the term's "plain meaning"; it failed to consider the specification *at all* in its claim-interpretation analysis other than to note the absence of an express definition. This demotion of the specification was also legal error.

In *Phillips*, this Court *en banc* explained the central role of the specification's disclosure in determining the broadest reasonable interpretation of claim language:

> The pertinence of the specification to claim construction is reinforced by the manner in which a patent is issued. The Patent and Trademark Office ("PTO") determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction "in light of the specification as it would be interpreted by one of ordinary skill in the art." *In re Am. Acad. of Sci. Tech. Ctr.,* 367 F.3d 1359, 1364 (Fed. Cir. 2004). Indeed, the rules of the PTO require that application claims must "conform to the invention as set forth in the remainder of the specification and the terms and phrases used in the claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description." 37 C.F.R. § 1.75(d)(1).

415 F.3d at 1316-17. *Accord Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 389 (1996) ("the standard construction rule that a term can be defined only in a way that comports with the instrument as a whole"); *U.S. v.*

Adams, 383 U.S. 39, 48-49 (1966) ("While the claims of a patent limit the invention, and specifications cannot be utilized to expand the patent monopoly, it is fundamental that claims are to be construed in the light of the specifications and both are to be read with a view to ascertaining the invention."); *Tempo Lighting,* 742 F.3d at 977 (affirming Board's BRI, in part as being supported by the specification, and rejecting the examiner's narrower interpretation based on dictionary definitions); *In re Suitco Surface, Inc.*, 603 F.3d 1255, 1260-61 (Fed. Cir. 2010) (reversing Board's claim interpretation for being unreasonably broad, and noting that the Board may not ignore "the specification and teachings in the underlying patent").

The MPEP likewise explains that a specification may inform a claim term's meaning *implicitly*: "The specification should also be relied on for more than just explicit lexicography or clear disavowal of claim scope to determine the meaning of a claim term when applicant acts as his or her own lexicographer; the meaning of a particular claim term may be defined by implication, that is, according to the usage of the term in the context in the specification." MPEP § 2111.01(IV).

The Board here, however, did not engage in this required analysis of the specification. Instead, it mistakenly reduced a patent specification to nothing more than a possible source of *express* definitions. Specifically,

Microsoft had pointed to several ways in which the specification's disclosure of the problem and purported solution, favored Microsoft's broader reading of "search." A00636-7. Microsoft explained that the narrower reading of "search" suggested by Proxyconn lacked any support in the specification. *Id.* Microsoft also explained that Proxyconn's own expert, Dr. Konchitsky, had agreed with Microsoft's analysis of the specification vis-à-vis "search." A00637. Yet the Board ignored all this in its BRI analysis. Instead, the one and only mention it made of the specification in its BRI analysis of "search" was the specification's lack of an express definition of "search." A00020. That demotion of the specification's role cannot be reconciled with this Court's precedents, the U.S. Supreme Court's precedents, or the MPEP.

F. The Board Committed Legal Error By Reading Into Claims A Limitation That Is Not Even Disclosed In The Patent

The Board's BRI appears to limit the "searching" step to an operation that compares the received digital digest to multiple digital digests at the receiver until a match is found. As Microsoft explained to the Board, however, the specification does not describe any embodiments that perform the "searching" step in the manner required by the Board's construction. That the Board's claim construction has "the effect of placing all the embodiments of the invention outside the scope of the claims is powerful

Case: 14-1542 Document: 12 Page: 39 Filed: 08/08/2014

evidence that [the] construction is incorrect." *Nellcor Puritan Bennett, Inc. v. Masimo Corp.,* 402 F.3d 1364, 1368 (Fed. Cir. 2005); *see Microsoft Corp. v. Multi-Tech Systems, Inc.,* 357 F.3d 1340, 1354 (Fed. Cir. 2004) "([The specification] consistently describes the hands-free interface as simply a microphone and a speaker. To require more structure would impermissibly exclude a preferred embodiment from the claim limitation…We therefore conclude that the term 'speaker phone' requires no physical structure beyond a microphone and a speaker.").

Nowhere does the specification *explicitly* disclose performing two or more digest comparison operations as part of checking (searching) for a matching digest value. Nor is that inherent (necessarily present). On the contrary, such an operation would be nonsensical in this context. Returning to the example of this Brief, the patent's algorithm would compare the remote and local digests (fingerprints) for this Brief and ask for an up-to-date copy from the network if there is a mismatch. It would make no sense to compare the Brief's digest with digests for other local documents such as a lunch menu.

Although the Board noted Microsoft's argument that the specification failed to disclose checking multiple digests during a "search" operation (A00019-20), it had no answer. Nowhere did the Board find that the

multiple-comparisons limitation it read into the claim term "search" was disclosed in the specification, either expressly or inherently.

G. The Board's Interpretation Is Unclear

Finally, the Board's capability-oriented interpretation of "searching"—"requires an ability to identify a particular data object with the same digital digest from a set of potentially many data objects stored in the network cache memory"—is unclear.

First, it conflates a method step, which is an action, with a capability, which is a function of a device. This is a method claim. "Searching" is an action, not a capability. Yet the Board converted this action, "searching," into a thing, "a 'search,'" and then restricted that action as requiring a capability it associated with that thing: "a 'search' for a desired member of a 'set' requires a *capability* to examine more than one item to identify a particular item within that set. Therefore, we conclude that 'searching for data with the same digital digest in said network cache memory' requires an *ability* to identify a particular data object with the same digital digest from a set of potentially many data objects stored in the network cache memory." A00020-21 (emphases added). But an action is not a function and a function is not an action. If, as here, there is only one potential candidate for a match—only one needle in the haystack, only one dog lost in the woods, or

only one set of car keys in the house—the act of looking for and finding that single item surely is the act of "searching," even if the person looking cannot distinguish one needle from another, one dog from another, or one car key from another.

To make matters worse, it is unclear what capability the Board has required. Does it require perfection—the ability to find a matching data object even if that data object has a different name or is otherwise an unlikely candidate? Does it require the ability to calculate or obtain a digest for every data object in the network cache (and comparing it to the received digest) before giving up and returning a negative indication signal? In view of the statutory mandate that patent claims "particularly point[ ] out and distinctly claim[ ]" the invention, 35 U.S.C. § 112, ¶ 2, an unclear interpretation is not a "reasonable" interpretation.

H. The Board's Error Affected Its Decision And Requires Remand

Microsoft contended that claim 24 is unpatentable for anticipation by Perlman and for obviousness over the combination of Perlman and Yohe. A00095. As noted, both Perlman and Yohe checked for a matching digest in the same manner as the alleged invention of the '717 patent. A00118; A00642-3, A00644. The Board initially found a reasonable likelihood that Microsoft would establish each of these grounds of unpatentability. A00352-

4. In its Final Written Decision, however, the Board rejected anticipation of claim 24 by Perlman (A00035-36), anticipation by Yohe of claim 22 (A00039-40), and obviousness of claim 24 over the combination (A00052-3), all on the same premise that a "search" requires more than a single comparison between a single pair of digests for the same data object.

As to Perlman, it reasoned:

> We conclude that Microsoft has failed to establish by a preponderance of evidence that Perlman meets the "searching" step recited in claim 22. Every portion of Perlman upon which Microsoft relies involves comparing one received digest to a corresponding locally calculated digest. These one-to-one comparisons cannot identify a particular data object from among a set of data objects as required by the "searching" step. Rather, the comparisons reveal whether a locally stored data object is synchronized with the corresponding remotely stored data object.

A00036.

As to Yohe, it concluded:

> Based on our review of Yohe, the directory signature comparator 46 performs, at most, a series of one-to-one comparisons of received digests to locally generated digests for directory sub-objects. Ex. 1005, col. 7, l. 6 – col. 8, l. 25 (describing steps performed in DIRECTORY REQUEST function as shown in Figures 15 and 16). We conclude that such one-to-one comparisons do not identify a particular directory sub-object from among a set of directory sub-objects as required by the "searching" step.

A00039.

And as to the combination, it stated that "[w]e cannot conclude that a claim would have been obvious when the prior art does not describe every element recited in the claim." A00052-53.

Because the Board's ruling that Microsoft failed to establish unpatentability depended on the Board's erroneous interpretation of the "searching" step, the Court should vacate that determination and remand for further proceedings under the correct BRI. *In re Abbott Diabetes Care Inc.*, 696 F.3d 1142, 1151 (Fed. Cir. 2012) (remanding for the Board to apply the correct BRI).

CONCLUSION

The Court should adopt Microsoft's proposed BRI and remand to the Board to use that BRI in its analysis of the patentability of claim 24.

Respectfully submitted.

/s/John D. Vandenberg
John D. Vandenberg
KLARQUIST SPARKMAN, LLP
One World Trade Center
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204-2988
(503) 595-5300
john.vandenberg@klarquist.com

Counsel for Appellant
Microsoft Corporation

August 8, 2014

ADDENDUM

TABLE OF CONTENTS

Trials@uspto.gov
Tel: 571-272-7822

Paper 73
Entered: February 19, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION
Petitioner

v.

PROXYCONN, INC.
Patent Owner

Case IPR2012-00026
Case IPR2013-00109
Patent 6,757,717

Before SALLY C. MEDLEY, THOMAS L. GIANNETTI, and MITCHELL G. WEATHERLY *Administrative Patent Judges.*

WEATHERLY, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. BACKGROUND

A. Introduction

On September 18, 2012, Microsoft Corporation ("Microsoft"), filed a petition under 35 U.S.C. §§ 311-319 for *inter partes* review of claims 1, 3, 10–12, 14, and 22–24 of U.S. Patent No. 6,757,717 ("the '717 Patent"). IPR2012-00026, Paper 6 ("the '026 Petition"). We granted the '026 Petition as to certain challenges to the patentability of claims 1, 3, 10, and 22–24, and denied the '026 Petition as to all challenges to the patentability of claims 11, 12, and 14 on December 21, 2012. IPR2012-00026, Paper 17 ("the '026 Decision").

Soon afterward, on January 11, 2013, Microsoft filed a second petition for *inter partes* review, this time challenging the patentability of claims 6, 7, 9, 11, 12, and 14 of the '717 Patent. IPR2013-00109, Paper 1 ("the '109 Petition"). Microsoft concurrently filed a motion to join IPR2013-00109 with IPR2012-00026. IPR2013-00109, Paper 7. We granted the '109 Petition as to certain challenges to patentability of claims 6, 7, 9, 11, 12, and 14 of the '717 Patent. IPR2013-00109, Paper 14 ("the '109 Decision"). We also granted Microsoft's motion for joinder and joined IPR2013-00109 with IPR2012-00026. IPR2013-00109, Paper 15.

After institution and joinder of both trials, Proxyconn, Inc. ("Proxyconn") filed its Corrected Patent Owner's Response ("Resp."). Paper 45.[1] Proxyconn also filed Patent Owner's Corrected Motion to Amend ("Mot. Amend") in which Proxyconn moved to substitute claims 35–

[1] This reference to "Paper" and all other references to "Paper" from this point forward in this Final Written Decision are to papers filed in the joined proceeding, which is captioned as IPR2012-00026 and IPR2013-00109.

41 for claims 1, 3, 6, 10, 11, 22, and 23, respectively, if the Board were to cancel any of those challenged claims as unpatentable. Paper 44.[2] This Final Written Decision addresses challenges to the patentability of claims 1, 3, 6, 7, 9–12, 14, and 22–24. Because claims 1, 3, 6, 10, 11, 22, and 23 are found unpatentable, this Decision also addresses the patentability of proposed substitute claims 35–41.

B. The '717 Patent

The '717 Patent describes a system for data access in a packet switched network. Ex. 1002, Abstract. The system has a sender/computer including an operating unit, a first memory, a permanent storage memory, and a processor. The system also has a remote receiver/computer including an operating unit, a first memory, a permanent storage memory, and a processor. The sender/computer and receiver/computer communicate through the network. *Id*. The sender/computer further includes a device for calculating digital digests on data; the receiver/computer further includes a network cache memory and a device for calculating digital digests on data in the network cache memory; and the receiver/computer and/or the sender/computer includes a device for comparison between digital digests. *Id.*

As described in the '026 Petition, the '717 Patent provides a way to reduce the amount of redundant data transmitted over a network. '026 Petition, 4. The processes described in the '717 Patent check for the identity

[2] Proxyconn filed Patent Owner's Motion to Amend under 37 C.F.R. § 42.121 on May 21, 2013. Paper 37. In an Order entered June 20, 2013, Proxyconn was granted permission to file its Corrected Motion to Amend to address typographical errors and file corrected exhibits. Paper 43. Proxyconn filed its Corrected Motion to Amend later that same day.

between two sets of data by comparing respective digital fingerprints of that data. *Id.* As described in the Summary of the Invention:

> If a sender/computer in the network is required to send data to another receiver/computer, and the receiver/computer has data with the same digital digest as that of the data to be sent, it can be assumed with sufficient probability for most practical applications that the receiver/computer has data which is exactly the same as the data being sent. Then, the receiver/computer can use the data immediately without its actual transfer through the network. In the present invention, this idea is used in a variety of ways.

Ex. 1002, col. 2, ll. 16-24.

The patent discloses several embodiments. In one, a sender/computer required to send data to a receiver/computer initially sends a digital digest of the data. If the receiver/computer already has data with the same digital digest, it uses this data as if it were actually transmitted from the sender/computer. *Id*. at col. 2, ll. 26-31. This embodiment is illustrated in Figures 5-7. Figure 5 is reproduced below:

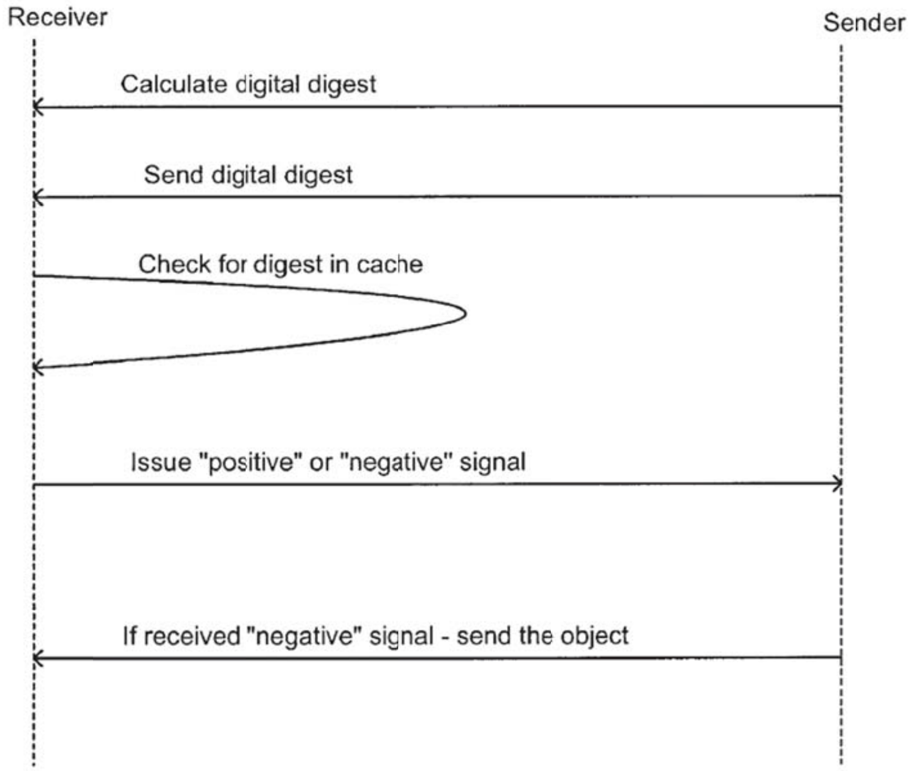


FIG. 5

Figure 5 is a schematic representation illustrating the interaction between a sender/computer and a receiver/computer according to the teachings of one embodiment of the ’717 Patent. *Id*. at col. 5, ll. 49-51.

In this embodiment, the receiver/computer receives a digital digest from a sender/computer and searches its network cache memory for data with the same digest. If the receiver/computer finds such data, it uses that data as if the data were received from the sender/computer and issues a positive indication signal to the sender/computer. Otherwise it sends a negative indication signal to the sender/computer. *Id*. at col. 7, ll. 51–60.

In another embodiment, auxiliary digital digests for other data objects can be sent together with the principal digest. If the receiver/computer cannot find data having the principal digest, it searches for data with one of the auxiliary digests. If such data is found, the sender/computer is required

to send only the difference between the requested data object and the data object corresponding to the auxiliary digest. *Id*. at col. 2, ll. 31-37. The expression in the Specification "difference between the first data or data object and the second data or data object" means any bit sequence that enables the restoration of the first data, given the second data, the bit sequence, and the method employed in calculating the difference. *Id*. at col. 2, ll. 38–42. This embodiment is illustrated in Figures 8-10. Figure 8 is reproduced below:

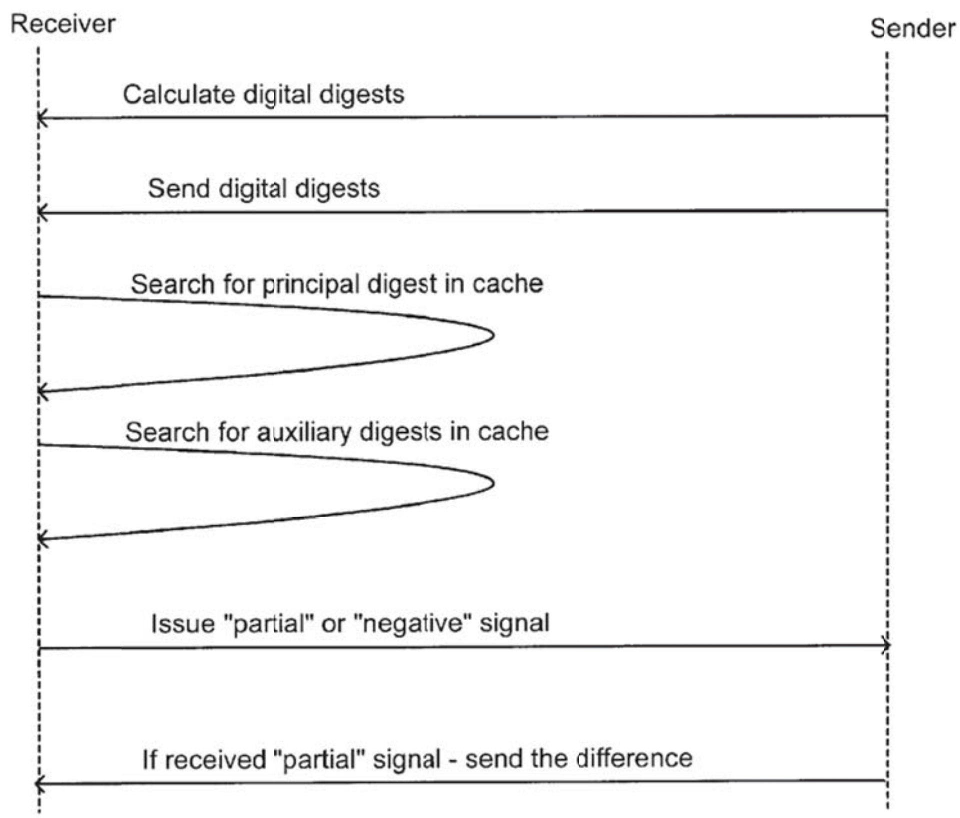


FIG. 8

Figure 8 is a schematic representation illustrating the interaction between a sender/computer and a receiver/computer according to the teachings of another embodiment of the invention. *Id*. at col. 5, ll. 59-61.

In this embodiment the sender/computer sends the principal and auxiliary (e.g., of a previous version of the data requested) digests to the receiver/computer. Upon receiving a message with these digital digests from the sender/computer, the receiver/computer searches its network cache memory for data having the same principal digest. If such data is found, the receiver/computer uses the data as if the data were received from the sender/computer and issues a positive indication signal to the sender/computer. Otherwise, the receiver/computer searches its network cache memory for data with the auxiliary digests. If it finds data with a digital digest substantially equal to one of the auxiliary digests, it issues a partial indication signal to the sender/computer, along with a reference to the digest. Otherwise it issues a negative indication signal to the sender/computer. *Id*. at col. 8, ll. 11-39.

C. Exemplary Claims

Claims 1, 6, 10, 11, and 22 are the independent claims among the challenged claims of the ’717 Patent. Claims 1, 6, and 10 are directed to systems, and claims 11 and 22 are directed to methods. The independent challenged claims, which are illustrative of the claims at issue in this *inter partes* review, recite:

> 1. A system for data access in a packet-switched network, comprising:
>
> a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, said sender/computer and said receiver/computer communicating through said network;

said sender/computer further including means for creating digital digests on data;

said receiver/computer further including a network cache memory and means for creating digital digests on data in said network cache memory; and

said receiver/computer including means for comparison between digital digests.

6. A system for data access in a packet-switched network, comprising:

a gateway including an operating unit, a memory and a processor connected to said packet-switched network in such a way that network packets sent between at least two other computers pass through it;

a caching computer connected to said gateway through a fast local network, wherein said caching computer includes an operating unit, a first memory, a permanent storage memory and a processor;

said caching computer further including a network cache memory in its permanent storage memory, means for calculating a digital digest and means for comparison between a digital digest on data in its network cache memory and a digital digest received from said packet-switched network through said gateway.

10. A system for data access in a packet-switched network, comprising:

a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, said sender/computer and said receiver/computer communicating through a network;

said sender/computer further including means for creating digital digests on data, and

said receiver/computer further including a network cache memory, means for storing a digital digest received from said network in its permanent storage memory and means for comparison between digital digests.

11. A method performed by a sender/computer in a packet-switched network for increasing data access, said sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and said sender/computer being operative to transmit data to a receiver/computer, the method comprising the steps of:

creating and transmitting a digital digest of said data from said sender/computer to said receiver/computer;

receiving a response signal from said receiver/computer at said sender/computer, said response signal containing a positive, partial or negative indication signal for said digital digest, and

if a negative indication signal is received, transmitting said data from said sender/computer to said receiver/computer.

22. A method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of:

receiving a message containing a digital digest from said network;

searching for data with the same digital digest in said network cache memory,

if data having the same digital digest as the digital digest received is not uncovered, forming a negative indication signal and transmitting it back through said network; and

creating a digital digest for data received from said network cache memory.

Ex. 1002, col. 10, l. 31 to col. 12, l. 45.

D. Remaining Challenges to the Patentability of Claims

We instituted this *inter partes* review in connection with the following challenges to the patentability of claims in the ’717 Patent:[3]

1. Anticipation by Perlman: claims 1, 3, and 22-24;
2. Anticipation by Yohe: claims 1, 3, 6, 7, 10, 22, and 23;
3. Anticipation by Santos: claims 1, 3, 10, 22, and 23;
4. Anticipation by DRP: claims 6, 7, 9, 11, 12, and 14;
5. Obviousness over the combination of Perlman and Yohe: claims 1, 3, 10, and 22-24; and
6. Obviousness over the combination of Mattis and DRP: claims 6, 7, 9, 11, 12, and 14.

’026 Decision 25–26; ’109 Decision 20.

II. ANALYSIS

A. Claim Interpretation

We interpret patent claim language in an *inter partes* review by ascribing to that language its broadest reasonable meaning in light of the specification of the patent. 37 C.F.R. § 42.100(b); Office Patent Trial

[3] The challenges to patentability are based upon five prior art references: US 5,742,820, issued Apr. 21, 1998 (Ex. 1003) (“Perlman”); US 5,835,943, issued Nov. 10, 1998 (Ex. 1005) (“Yohe”); Santos and Wetherall, INCREASING EFFECTIVE LINK BANDWIDTH BY SUPPRESSING REPLICATED DATA (June 1998) (Ex. 1004) (“Santos”); THE HTTP DISTRIBUTION AND REPLICATION PROTOCOL, W3C Note (August 25, 1997), retrieved from http://www.www3.org/TR/NOTE-drp-19970825 (IPR2013-00109, Ex. 1003) (“DRP”); US 6,292,880 B1, issued Sep. 18, 2001 (IPR2013-00109, Ex. 1004) (“Mattis”).

Practice Guide, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012). We also interpret claim language according to its ordinary and customary meaning to one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007).

We expressly interpret below only those claim terms that require analysis to resolve arguments related to the patentability of the challenged claims. Except as otherwise stated, we interpret the remaining claim terms as set forth in the ’026 Decision and the ’109 Decision.

1. Data Access

Each contested claim recites “data access.” Ex. 1002, col. 10, l. 31 (claims 1, 3), col. 10, l. 64 (claims 6, 7, and 9), col. 11, l. 20 (claim 10), col. 11, l. 35 (claims 11, 12, and 14), col. 12, l. 30 (claims 22–24). Proxyconn urges that “data access” means “obtaining data . . . on a remote computer on a network, in response to a request from a client.” Resp. 11 (citing Ex. 1002, col. 1, ll. 18–26; *id. at* col. 7, ll. 65–67). In support, Proxyconn cites portions of the Specification of the ’717 Patent that describe exemplary data transmission sessions in which a network client “requests” data from a server. The first cited portion describes such interactions between a client and server as “prior art.” Ex. 1002, col. 1, ll. 18–26. The second cited portion states: “This transaction begins with a receiver/computer sending a request to the sender/computer.” *Id*. at col. 7, ll. 65–67. The phrase “[t]his transaction” refers to the interaction between the receiver/computer and the sender computer depicted in Figures 5–7 of the ’717 Patent. *Id*. at col. 7, ll. 51–67.

By contrast, Microsoft contends that “data access” means “data acquisition.” Microsoft Corporation’s Reply to Patent Owner’s Corrected

Response ("MS Reply"), 2 (Paper 46). Microsoft dismisses the portions of the Specification that Proxyconn cites as neither mentioning "data access" nor narrowly defining "data access." *Id*. Microsoft argues that other portions of the Specification imply that the step of the receiver/computer requesting data is merely optional. *Id*. (citing Ex. 1002, 8:37–39). Proxyconn's expert, Dr. Konchitsky, testified that the Specification describes scenarios in which a sender transmits data to a receiver without a request from the receiver. *See* Ex. 1024, 69:1–24, 71:8–22 (describing the data communication method illustrated in Figure 8 of the '717 Patent). Microsoft also points out that claim 32, which is not challenged, explicitly recites a method in which a client sends a request for data to a server. Paper 72, Final Hearing Transcript 10:9-12, 79:22–80:9 ("Tr.").

Both parties' interpretations of "data access" are too narrow. Neither the challenged claims nor the Specification expressly limits "data access" to require a "request from the client" as proposed by Proxyconn. The claims merely recite "data access." Even though the Specification describes examples in which the client requests data from a server, the Specification does not require that the client request data in all described embodiments of the claimed systems and methods. For example, the Specification expressly describes an embodiment in which "a sender/computer required to send data to a receiver/computer . . . initially sends a digital digest of the data." Ex. 1002, col. 2, ll. 26–28. "[L]imitations are not to be read into the claims from the specification." *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (citing *In re Zletz*, 893 F.2d 319, 321(Fed. Cir. 1989)). We decline to do so here.

Microsoft's position is similarly unsupported by the claims themselves or the Specification. None of the challenged independent claims affirmatively recites that the receiver/computer *acquires* data from the sender/computer. Microsoft cites no portion of the Specification, and we find no support for the proposition that the Specification equates "data access" with "data acquisition."

We determine that the plain meaning of "data access" is clear. Independent challenged claims 1, 6, 10, 11, and 22 recite "access" as a noun modified by "data." Ex. 1002, col. 10, l. 31 (claims 1, 3), col. 10, l. 64 (claims 6, 7, and 9), col. 11, l. 20 (claim 10), col. 11, l. 35 (claims 11, 12, and 14), col. 12, l. 30 (claims 22–24). "Access" plainly means the "freedom or ability to obtain or make use of." MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 6 (10th ed. 1999). We conclude, therefore, that the claimed systems and methods recite "data access" to refer to the freedom or ability to obtain or use data. Although obtaining or acquiring data requires access to that data, access to the data need not involve acquisition of that data.

2. Permanent Storage Memory

Claims 1, 3, 6, 7, 9, 10, and 22–24 recite "permanent storage memory." Ex. 1002, col. 10, l. 31 – col. 13, l. 8. Proxyconn argues that "permanent storage memory" means non-volatile memory that can be used for writing and reading data and does not refer to read-only memory ("ROM"). Resp. 12. The Specification states "an example of a permanent storage memory may be a disk drive, a flash RAM or a bubble memory." Ex. 1002, col. 7, ll. 38–40. In support of its proffered definition of "permanent storage memory," Proxyconn also cites Yohe's statement that "'[p]ermanent storage memory,' as used herein, includes but is not limited

to, disk drive, flash RAM or bubble memory, for example." Resp. 12 (quoting Ex. 1005, col. 3, ll. 5–7).

Microsoft counters that "permanent storage memory" is not restricted to non-volatile memory that permits multiple write operations, but may also include storage that is write-once, read-many ("WORM") memory. MS Reply 2. Microsoft contends that a CD optical storage disc, a type of non-volatile WORM memory, would constitute permanent storage memory. *See id*. (citing Ex. 1024, 88:7–89:12). Thus, the dispute centers on whether "permanent storage memory" encompasses ROM and other types of WORM types of non-volatile memory.

The testimony of both experts persuades us that a skilled artisan would interpret "permanent storage memory" to cover non-volatile memory that supports multiple write operations. Dr. Long equated the "permanent storage" described in the '717 Patent with a "disk" or "flash" memory. Ex. 1026, 97:15–98:10. Dr. Konchitsky testified that a skilled artisan would have considered "permanent storage memory," which enables writing or storing of information, to differ from "permanent memory," which can only be read after being written one time "in factory." Ex. 2002 ¶ 21. The ability to write data many times to permanent storage memory is consistent with the way that "permanent storage memory" is used in the context of at least claim 6. Claim 6 recites a "caching computer further including a network cache memory in its permanent storage memory." The presence of cache memory, which is likely to be written many times, in the "permanent storage memory" implies a capability to write data many times to the claimed "permanent storage memory." Because claim terms are normally used consistently throughout the patent, the usage of a term in one claim may

illuminate the meaning of the same term in other claims. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005). Therefore, we interpret "permanent storage memory" to mean any non-volatile memory that supports multiple write operations.

3. Sender/Computer and Receiver/Computer

Challenged claims 1, 3, 10, 11, 12, 14, and 22–24 recite either a "sender/computer" or "receiver/computer" or both. Ex. 1002, col. 10, l. 31 – col. 13, l. 8. Previously, we interpreted "sender/computer" to mean a computer that sends data and "receiver/computer" to mean a computer that receives data. '026 Decision 14. We also concluded that each of these respective computers can encompass multiple devices including intermediaries. *Id.*

Proxyconn argues that our interpretation is "inconsistent with the '717 Patent, is not the broadest *reasonable* interpretation of the claim terms, and should be revised to exclude separate intermediate computers such as gateways, proxies, routers, and caching computers." Resp. 13. Proxyconn contends that the Specification consistently refers to the sender and receiver computers as separate devices.

Microsoft contends that we correctly interpreted the computers to encompass multiple devices including intermediate devices. The Specification represents the receiver and sender computers (46, 42 respectively) in decidedly schematic form, as shown in Figures 4, 11, and 14, reproduced below.

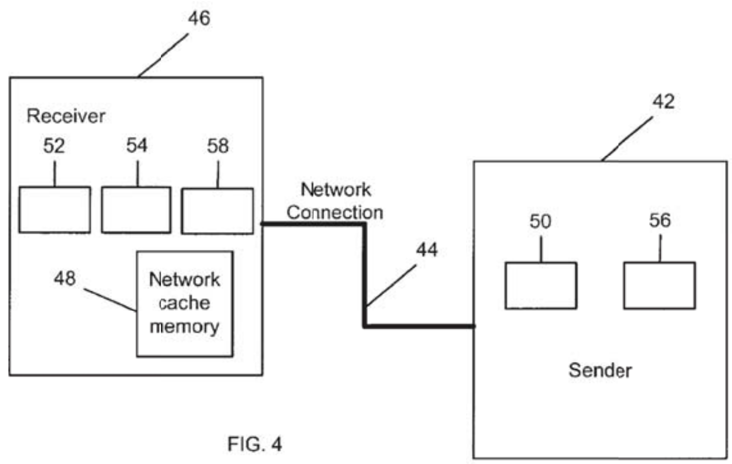


FIG. 4

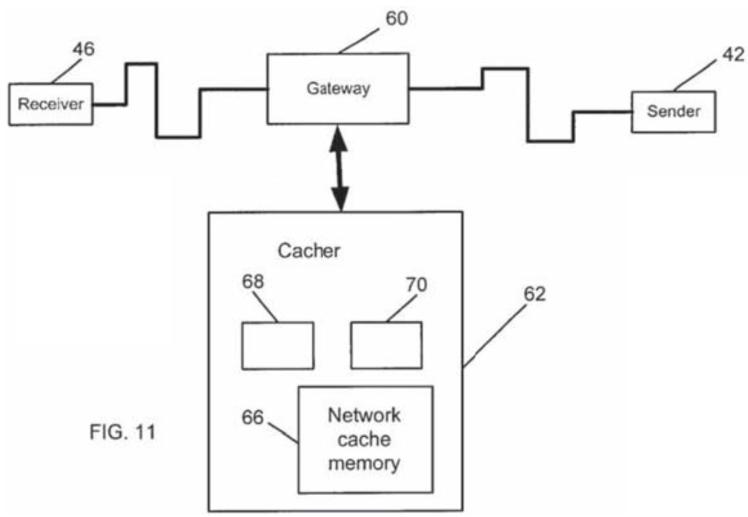


FIG. 11

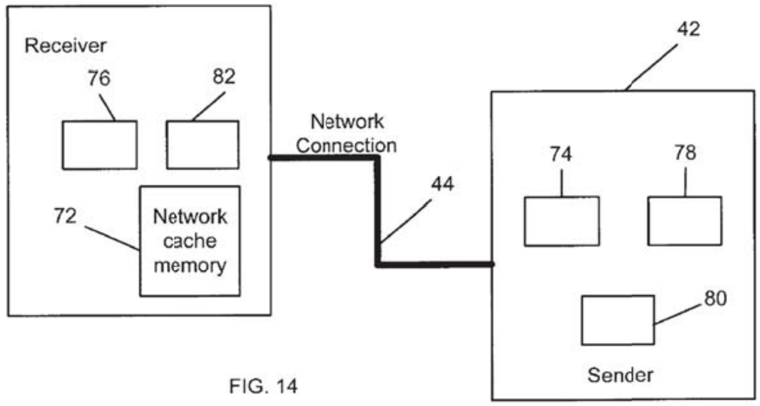


FIG. 14

Figures 4, 11, and 14, reproduced from top to bottom above, schematically illustrate the claimed receiver/computer and sender/computer in various network configurations as functional block diagrams.

Case: 14-1542 Document: 12 Page: 62 Filed: 08/08/2014

Figure 4 illustrates receiver/computer 46 as a collection of functionally defined subsystems 48, 52, and 54, which are described as follows: "[T]he receiver/computer has calculating means 52 for calculating a digital digest on data stored in its network cache memory 48. The receiver/computer also has comparison means 54 for comparing between such a calculated digital digest and a digital digest received from the network." Ex. 1002, col. 7, ll. 32–37.

The Specification appears to have one instance in which a computer is described as being separate from or integral with another computer. The Specification implies that gateway 60 and caching computer 62 may be separate devices, but only by noting that "gateway computer **60** may be integrally formed with the caching computer." Ex. 1002, col. 9, ll. 6–8. The Specification, along with the above figures, conveys to a skilled artisan that the described computers, including the receiver/computer and the sender/computer, may or may not be located in separate housings. Accordingly, Proxyconn has not persuaded us to modify the original interpretation of "receiver/computer" and "sender/computer."

4. Gateway . . . Between at Least Two Other Computers

Independent claims 6 recites a "gateway . . . connected to said packet-switched network in such a way that network packets sent between at least two other computers pass through it." Ex. 1002, col. 10, l. 66 – col. 11, l. 2. Claims 7 and 9, which depend upon claim 6, also include the "gateway" and "two other computers." *Id*. at col. 11, ll. 13–20. Proxyconn asserts that "two other computers" refers to "the sender/computer and the receiver/computer." Resp. 15 (citing Ex. 1002, col. 2, ll. 44–47). The cited portion of the Specification, however, merely recites verbatim the language

of claim 6 relating to the gateway and two other computers. Therefore, the cited portion has not been shown to support Proxyconn's contention. Microsoft contends that no such limitation exists on the "two other computers" and that these computers may be *any* two other computers connected on the network to the gateway. *See* '109 Petition 13–14, Appendix A 5–6.

We agree with Microsoft. Claim 6 plainly and unambiguously recites "two other computers" as a limitation on the manner in which the "gateway" is "connected to said packet switched network." That is, the gateway is connected to the network so that "network packets sent between at least two other computers pass through it [i.e., the gateway]." Applying the broadest reasonable interpretation, we conclude that claim 6 does not limit which computers may constitute the "two other computers" between which the gateway is connected.

5. *Means for comparison between digital digests*

a. Claims 1 and 3

Claim 1 recites "means for comparison between digital digests." Resolution of the parties' arguments relating to whether Yohe anticipates claims 1, 3, and 10 requires that we interpret "digital digests" as recited in the comparison means. We interpret "digital digests" by reading claim 1 in its entirety. Claim 1 recites that both the sender and receiver include "means for creating digital digests on data." We conclude that the "digital digests" recited in the means for comparison refers to the "digital digests on data" that are recited earlier in claim 1 in the "means for creating."

b. Claim 10

Claim 10, like claim 1, recites that the receiver includes "means for comparison between digital digests." Also like claim 1, claim 10 recites that the sender includes a "means for creating digital digests on data." By contrast to claim 1, claim 10 does not recite a "means for creating" in the receiver. Ex. 1002, col. 11, ll. 20–33. Instead, the receiver includes a "means for storing *a digital digest* received from said network." *Id.* at col. 11, ll. 31–32 (emphasis added). The reference to "a digital digest" rather than "the digital digest on data" in the storing means implies that the receiver can store any type of digital digest received from the network.

Therefore, the "digests" that are compared in the "means for comparison" recited in claim 10 need not be the two digests on data created by the sender and receiver. Instead, the "means for comparison between digital digests" recited in claim 10 refers to structure that can compare any digital digest received from the network with any other digital digest.

6. Searching for Data with the Same Digital Digest

Claims 22–24 recite a step of "searching for data with the same digital digest." Proxyconn argues that the "searching" step requires the capability to identify particular data "with the same digital digest" from among a set of data that potentially contains multiple items. *See* Resp. 6, 20–21 (attempting to distinguish claims 22–24 from Perlman), 27–28 (attempting to distinguish claims 22 and 23 from Yohe), 35–36 (attempting to distinguish claim 23 from Santos). Microsoft contends that the '717 Patent equates "search" with "check for" and that the Specification never describes any "search" method other than "comparing two digest values for a match." MS Reply 4. Microsoft asserts that the recited step of "searching for data with the same

digital digest," merely requires comparing a digest for a data object received from the network with a digest of the receiver's copy of that data object. *Id.* at 5.

The Specification never expressly defines "search." Nonetheless, the plain meaning of "search" is: "to look into or over carefully or thoroughly in an effort to find or discover something." MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 1053 (10th ed. 1999). Two dictionaries in the relevant field of computing technology define "search" as it would be understood by a skilled artisan as follows:

1. "To scan one or more data elements of a set in order to find elements that have a certain property," IBM DICTIONARY OF COMPUTING 600 (10th ed. 1993); and
2. "(information processing). To examine a set of items for those that have a desired property," IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONICS TERMS 808 (3d ed. 1984).

These dictionary definitions reflect that a skilled artisan would have understood "search" to involve analyzing a set of items to identify one particular item from among a set of items. A "set" refers to "a number of things of the same kind that belong or are used together," MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 1071 (10th ed. 1999), or "[a] finite or infinite number of objects of any kind, of entities, or of concepts that have a given property or properties in common," IBM DICTIONARY OF COMPUTING 618 (10th ed. 1993). While a set can contain one item, a "search" for a desired member of a "set" requires a capability to examine more than one item to identify a particular item within that set. Therefore, we conclude that "searching for data with the same digital digest in said

network cache memory" requires an ability to identify a particular data object with the same digital digest from a set of potentially many data objects stored in the network cache memory.

B. The Prior Art

1. Perlman

Perlman generally relates to synchronization of information across a computer network. Ex. 1003, col. 1, ll. 6–8. Perlman's Figure 2 (reproduced below) is a block diagram of two computer networks to which multiple nodes, which include routers R1–R6, source nodes S1–S6, and a destination node D, are connected.

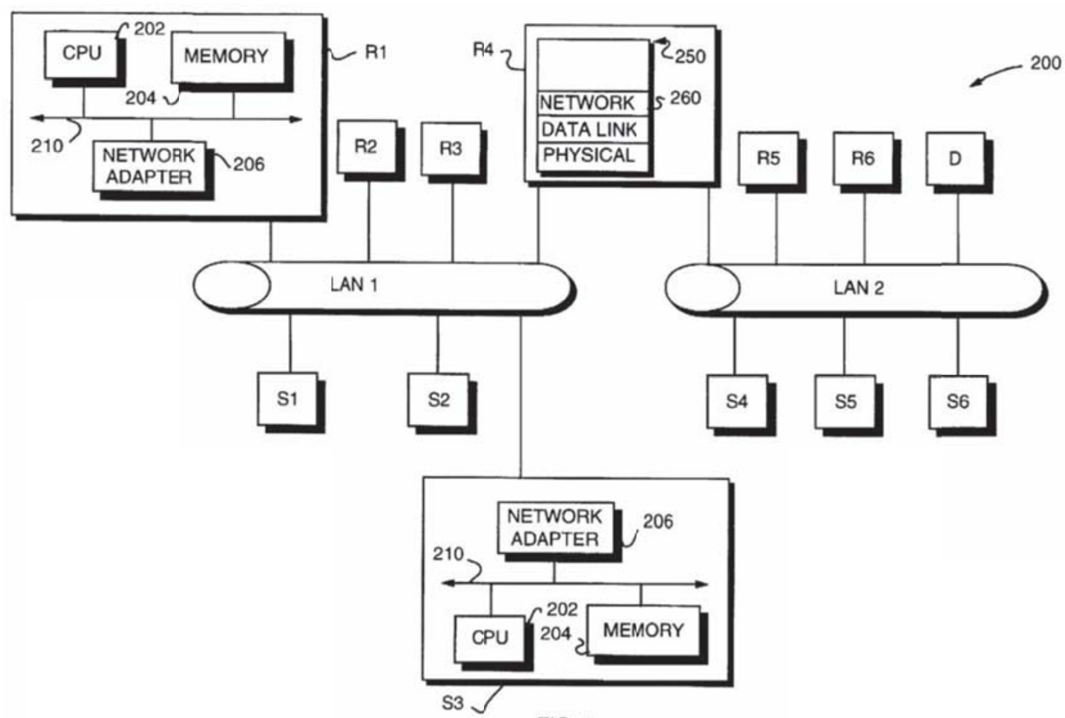


FIG. 2

Perlman's Figure 2 is a block diagram of two computer networks to which multiple nodes are connected.

Perlman states that the "nodes are typically general-purpose computers" and that "[e]ach node typically comprises a . . . memory unit **204**" which may

include "storage locations typically composed of random access memory (RAM) devices." *Id*. at col. 5, ll. 40–47.

Packetized data is transmitted across the network with each packet having the address of its final destination and the address of the next node to which it will travel along the route to the final destination. *Id*. at col. 5, l. 65 – col. 6., l. 1. The final destination address remains constant, but the "next destination" address changes as the packet moves from node to node in the network. *Id*. at col. 6, ll. 1–4. Upon arrival of a packet to a router, the router determines the next destination address of the packet based on algorithms to calculate a path to the final destination. *Id*. at col. 6, ll. 5–24. For this mode of transmission to work, every router must determine and communicate its location in the network to other nodes on the network. *Id*. at col. 6, ll. 25–45. These network "maps" must be synchronized to ensure that data packets arrive at the correct final destination. *Id*. at col. 6, ll. 46–53.

Perlman synchronizes these network maps by having one designated router (e.g., R4) periodically calculate and send a digest of that map (called a "complete sequence numbers packet" or CSNP) to all other routers on the network. *Id*. at col. 6, l. 47 – col. 7, l. 55. When each of the other routers receives the CSNP digest from R4, each of those routers compares that received CSNP digest to a digest of the network topology calculated locally by the receiving router. *Id*. at col. 7, ll. 56–63.

Perlman also describes an "alternate embodiment" in which "high-level and low-level identifiers are bundled within the same hello message that is periodically broadcast by the designated router, **R4** to the other routers." *Id*. at col. 8, ll. 25–28. For this embodiment, the receiving router first compares its locally generated high-level digest with received high-

level digest 710. *Id*. at col. 8, ll. 32–35. If the two digests are not the same, then the receiving router calculates low-level digests for fragments of its database and compares each of these low-level digests with corresponding received low-level digests 725a–c. *Id*. at col. 8, ll. 36–39. Based on these comparisons, the receiving router determines which fragments of its database require updating. *Id*. at col. 8, ll. 39–42.

2. Yohe

Yohe generally describes an "apparatus for increased data access in a network [that] includes a file server computer having a permanent storage memory, [and] a cache verifying computer operably connected to the file server computer in a manner to form a network for rapidly transferring data." Ex. 1005, Abstract. Yohe's Figure 2, reproduced below, schematically illustrates the configuration of the data access network.

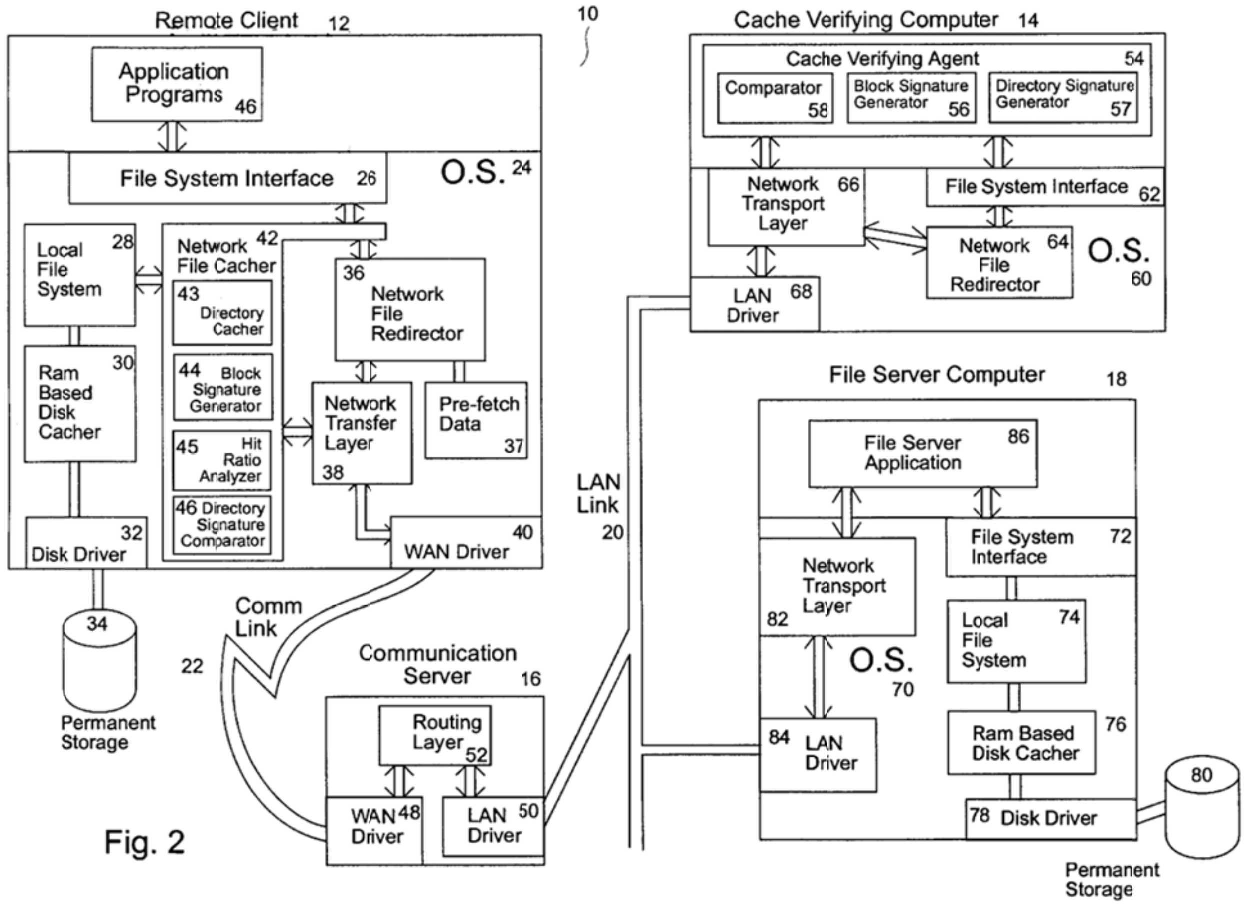


Fig. 2

Yohe's Figure 2 is a block diagram of a computer network including a remote client 12, cache verifying computer 14, communication server 16, and file server computer 18 that are connected to each other via either a LAN or WAN.

Yohe's apparatus reduces the time required for a remote client to access data on a file server using a caching computer and caching technique. *Id.* at col. 4, ll. 27–31.

Yohe's remote client computer 12 expressly includes a processor, operating system, permanent storage memory, and other memory. *Id.* at col. 2, ll. 51–54. The remote client computer also includes network file cacher 42 with block signature generator 44 and directory signature comparator 46. *Id.* at col. 5, ll. 1–6.

Similarly, Yohe's file server computer also expressly includes a processor, operating system, memory, and permanent storage memory. *Id.* at col. 3, ll. 22–24. The file server computer is "operably connected" to cache verifying computer 14, which includes "means for performing an operation on data stored in the permanent storage memory of the file server computer to produce a signature of the data characteristic of one of a file and directory." *Id.* at col. 2, ll. 43–51. Yohe describes two means for producing a signature, or digest, block signature generator 56 and directory signature generator 57. *Id.* at col. 5, ll. 14–17.

Yohe describes cache verifying computer 14 as having an operating system, a first memory, and a processor. *Id.* at col. 2, ll. 46–47. Cache verifying computer 14 incorporates cache verifying agent 54 consisting of block signature generator 56, directory signature generator 57, and comparator 58. *Id.* at col. 5, ll. 14–17. The cache verifying computer is also "operably connected" to the file server computer so that block signature generator 56 can create digests for data files stored in permanent storage 80. *Id.* at col. 2, ll. 43–51.

Yohe's client computer and the cache verifying computer thus each have capability to generate digests for data stored on the client and file server respectively. When Yohe's client computer requests data (e.g., to read a file stored in permanent storage on the file server), it generates a read request with an embedded digest and sends that request to the cache verifying computer. The cache verifying computer generates a digest for the requested file as that file exists in the permanent storage memory of the file server. The cache verifying computer then compares the two digests to determine whether the files stored on the remote client and file server are the

same. If not, the file server's version is sent to the remote client. If so, the remote client uses its locally stored version of the file. *Id*. at col. 6, ll. 22–37; figs. 6 and 7.

Yohe does not compare any particular received digest with more than one locally generated digest. Yohe describes two types of comparators for analyzing two versions of a digest. The first is comparator 58 in cache verifying computer 14, which compares digests for data files. *Id*. The second is directory signature comparator 46 in remote client 12, which compares one-by-one a series of digests for directory sub-objects that are received from the cache verifying computer with the locally generated digests for corresponding directory sub-objects. *Id.* at col. 7, l. 6 – col. 8, l. 25 (describing steps performed in DIRECTORY REQUEST function as shown in Figures 15 and 16).

3. Santos

Santos describes a compression architecture that prevents transmission of replicated data to increase bandwidth in a packet switched environment such as the Internet. Ex. 1004, 2. The bandwidth savings is achieved by transmitting repeated data as a short dictionary token, using caches of recently-seen data at both ends of the link to maintain the dictionary, and encode and decode the tokens. *Id.* at 5. The approach of Santos is based on the insight that the "fingerprint" of a data segment is an inexpensive name for the data itself, both in terms of space and time. *Id.* Santos uses the MD5 hash algorithm for his implementation, but states that other "fingerprints" could be used. *Id.* Figure 4 of Santos is reproduced below:

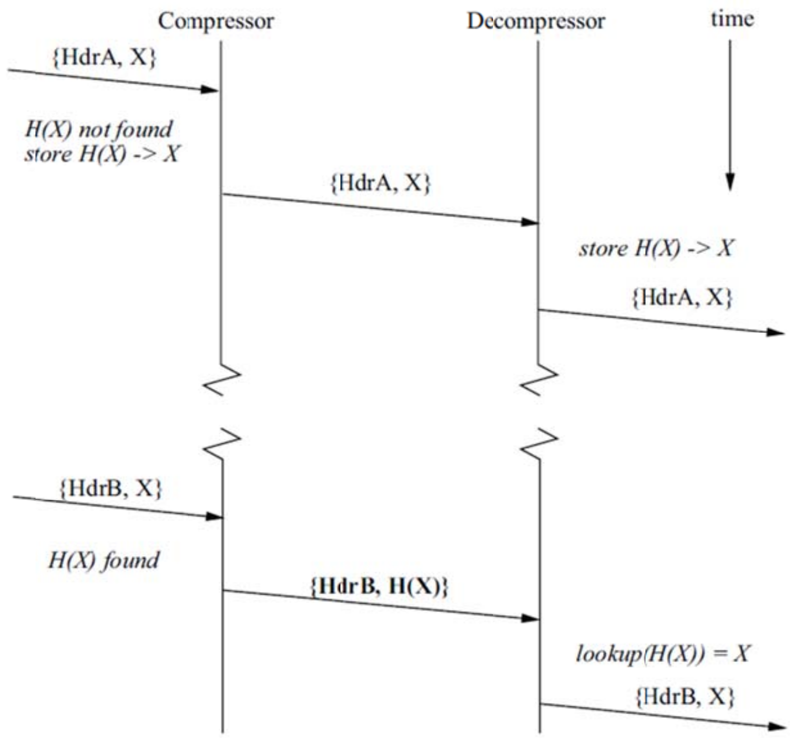


Figure 4: Compression protocol

Figure 4 of Santos shows a message exchange sequence from a sender (compressor) to a receiver (decompressor). Ex. 1004, 7.

The upper portion of the figure illustrates the sequence of events when the compressor receives a packet having header HdrA whose fingerprint H(X) is not in the cache. When this occurs, the compressor stores packet contents X in its cache, indexed by its fingerprint H(X), and forwards the header and contents across the link. The lower portion of the figure illustrates the sequence of events occurring when the compressor receives a packet having header HdrB and a fingerprint H(X) that is found in the cache. *Id*. at 7-8. When this occurs, the compressor sends the header and fingerprint, thus achieving a savings in bandwidth. *Id.* at 8. Santos implements the compressor and decompressor as two Pentium II based

machines with 128 MB of RAM and a cache of 200 MB running a Linux operating system. *Id*. at 9.

4. DRP

DRP describes a protocol for improving the efficiency and reliability of data distribution over HTTP. DRP, 2, ll. 12–13. One of the goals is to avoid downloading the same data more than once. *Id*. at 2, ll. 29–30. The protocol described in DRP makes use of content identifiers based on checksum technology. *Id*. at 2, ll. 37–39. A content identifier can be used uniquely to identify each piece of data or content and to determine whether two pieces of content are identical. An example of a checksum algorithm that can be used for this purpose is the MD5 message digest algorithm. *Id.* at 3, ll. 24–25.

To describe the exact state of a set of data files, DRP uses a data structure called an index. *Id.* at 4, l. 37. An index is a snapshot of the state of a set of files at a particular moment in time. It is typically stored in memory as a data tree structure, but to enable clients and servers to communicate this information over HTTP, an index can be described using XML. *Id*. at 4, ll. 39–42.

A DRP index is retrieved by giving a uniform resource locator ("URL") to the index. *Id*. at 5, l. 22. The index can be stored in any file and can be retrieved using a normal HTTP GET request. *Id*. at 5, ll. 22–23. Once the initial download is complete, a client can update content by downloading a new version of the index and comparing it against the previous versions of the index. Because each file entry in the index has a content identifier, the client can determine which files have changed and,

thus, determine the minimal set of files that need to be downloaded in order to bring the client up to date. *Id.* at 5, ll. 31–33.

An HTTP header field called Content-ID is used to specify the current correct version of the file that is requested by the client. The server can use the content identifier in the Content-ID field to determine if the requested version of the file can be delivered to the client. *Id.* at 7, ll. 30–32. If no content identifier is specified in the HTTP GET request, the server returns the current version of the file. *Id.* at 7, ll. 37–38. When a file is updated on the server, it will be downloaded by each of the clients that needs the new version. *Id.* at 8, ll. 3–4.

DRP notes that updates to files very often affect only small portions of the file, and it would therefore be much more efficient if the server could reply with only the parts of the file that have changed. *Id.* at 8, ll. 4–5. This is achieved using a "differential" GET request. When a file is modified, the client can issue a differential GET request for the file, which includes not only the content identifier of the desired version of the file, but also the content identifier of the current version of the file on the client. *Id.* at 8, ll. 11–13. In a differential GET request the content identifier of the file as it exists on the client is specified using the Differential–ID field in the HTTP header. *Id.* at 8, ll. 14–15. When the server receives a GET request that includes a Differential–ID field, it can look in its file cache for both versions of the requested file, using the content identifiers specified in the Content–ID field and the Differential–ID field. If both versions of the file are found, the server can compute the difference between the two files and return the difference, rather than the entire file. *Id.* at 8, ll. 25–28. If the server does not have access to the version of the file that is indicated by the differential

content identifier, it can ignore the differential content identifier and return the entire requested file. *Id*. at 8, ll. 32–33.

DRP also describes the use of proxy caching. In this application, an HTTP proxy can be made aware of the Content–ID and Differential–ID fields in HTTP requests and replies. *Id*. at 9, ll. 38–39. Because the content identifier is included in each GET request, the proxy can avoid accidentally returning the wrong version of the requested file. *Id*. at 9, ll. 39–40. The proxy can use the content identifier field to identify uniquely the content being transferred as the same content is likely to have the same content identifier, even when downloaded from multiple locations. The proxy can thus use this information to avoid multiple downloads. *Id*. at 9, ll. 43–45. The proxy can also use the Differential–ID header field to reply to differential GET requests. If both versions of the file are in the proxy's cache, the proxy can provide the differential reply. *Id*. at 10, ll. 1–2.

5. Mattis

According to Mattis, a key factor limiting the performance of the World Wide Web is the speed with which servers can supply information to clients via the Internet. Mattis, col. 1, ll. 53–55. Accordingly, client transaction time can be reduced by storing replicas of popular information objects in repositories geographically dispersed from the server. Each local repository for object replicas is generally referred to as a cache. *Id*. at col. 1, ll. 58–62. In some arrangements, the cache is located in a proxy server that is logically interposed between clients and the server. The proxy server is a "middleman gateway," acting as a server to the client, and the client to the server. *Id*. at col. 1. 66 to col. 2, l. 3. A proxy server equipped with a cache

is called a "caching proxy server," or just a "proxy cache." *Id*. at col. 2, ll. 3–5.

The proxy cache intercepts requests for resources that are directed from clients to the server. When the cache in the proxy has a replica of the requested resource that meet certain constraints, the proxy responds to the clients and serves the resources directly. *Id*. at col 2, ll. 6–11. In this arrangement, the number and volume of data transfers along the links are greatly reduced. As a result, network resources or objects are provided more rapidly to the clients. *Id*. at col. 2, ll. 11–14.

Mattis uses a "fingerprint" of the content that makes up the object itself to locate the object. *Id*. at col. 8, ll. 18–21. Specifically, the object key is a unique "fingerprint" or compressed representation of the contents of the object. A copy of the object is provided as an input to a hash function (e.g., MD5) and its output is the object key. Given a content fingerprint key, the content can easily be found in the cache. *Id*. at col. 8, ll. 23–36. In some embodiments of Mattis, for each of the objects, the cache also creates a name object key. The name key is created by applying a hash function to the name of the object. *Id*. at col. 8, ll. 55–58. Mattis recognizes that requests for objects typically identify requested objects by name, such as a URL, file system name, or network address. *Id*. at col. 9, l. 65 to col. 10, l. 4.

In one embodiment of Mattis, a lookup operation is used to determine whether a particular object identified by particular name is stored currently in the cache. *See id*. at fig. 9A. When the process is applied in the context of the World Wide Web, the name is a URL. *Id*. at col. 27, ll. 61–62. The cache converts the name of the object to a key value by passing the object

name or URL to hash function such as MD5. *Id*. at col. 27, ll. 63–67. The key is checked against a directory, and if the requested object is found, it is retrieved from storage and sent to the client. *Id*. at col. 28, ll. 10–14. If the object is not found in storage, the cache obtains a copy of the object from the appropriate server. *Id*. at col. 28, ll. 43–47.

C. Patentability of Original Claims

To prevail in its challenges to the patentability of claims, the petitioner must establish facts supporting its challenges by a preponderance of the evidence. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d).

1. Anticipation

The Court of Appeals for the Federal Circuit summarized the analytical framework for determining whether prior art anticipates a claim as follows:

> If the claimed invention was "described in a printed publication" either before the date of invention, 35 U.S.C. § 102(a), or more than one year before the U.S. patent application was filed, 35 U.S.C. § 102(b), then that prior art anticipates the patent. Although § 102 refers to "the invention" generally, the anticipation inquiry proceeds on a claim-by-claim basis. *See Hakim v. Cannon Avent Group, PLC*, 479 F.3d 1313, 1319 (Fed. Cir. 2007). To anticipate a claim, a single prior art reference must expressly or inherently disclose each claim limitation. *Celeritas Techs., Ltd. v. Rockwell Int'l Corp.*, 150 F.3d 1354, 1361 (Fed. Cir. 1998). But disclosure of each element is not quite enough—this court has long held that "[a]nticipation requires the presence in a single prior art disclosure of all elements of a claimed invention *arranged as in the claim*." *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 1548 (Fed. Cir. 1983) (citing *Soundscriber Corp. v. United States*, 175 Ct.Cl. 644, 360 F.2d 954, 960 (1966) (emphasis added)).

Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323, 1334–35 (Fed. Cir. 2008). We must analyze prior art references as a skilled artisan would. *See Scripps Clinic & Res. Found. v. Genentech, Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991) (to anticipate, "[t]here must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention").

For the reasons expressed below, Microsoft has demonstrated by a preponderance of evidence that Yohe anticipates claim 10; Santos anticipates claims 1, 3, 22, and 23; and DRP anticipates claims 6, 7, 9, 11, 12, and 14. Microsoft has failed to establish by a preponderance of evidence that Perlman and Yohe anticipate any of the remaining challenged claims.

a. Perlman

(1) Claims 1 and 3

Proxyconn argues that Perlman does not anticipate claims 1 and 3 because Perlman fails to disclose "permanent storage memory," but instead uses random access memory ("RAM"). Resp. 18 (citing Ex. 1003, col. 5, ll. 46–48)). Microsoft counters that Perlman describes permanent storage memory in the form of memory for storing an operating system. '026 Petition 17 (citing Ex. 1002, col. 5, ll. 41–52, fig. 2). We see no express statement in the cited portion of Perlman that its computers include any type of memory other than RAM. '026 Petition 17 (citing Ex. 1007, 12:5–17). Microsoft relies upon the Declaration of Darrell D. E. Long, Ph.D. Dr. Long opines, without explanation, that a skilled artisan would understand Perlman to disclose an "illustrative-embodiment router [that] is a 'general-purpose' computer . . . having, among other things, a hard disk or the like for storing

persistently data and application programs." Ex. 1007, 12:7–12 (citing Perlman, Ex. 1003, col. 5:41–52, fig. 2). However, Dr. Long also testified:

> Q. So your testimony is that each and every limitation of claim one is disclosed in Perlman, correct?
>
> A. We've got a computer — there may be — I'd have to go back and study Perlman. Maybe we don't mention permanent storage, whatever that means.
>
> Q. Okay.
>
> A. Okay. Although, certainly that's — I would consider that a triviality.
>
> Q. Okay.
>
> A. Okay. Digest, that's there. That's there. Comparisons are there. Maybe permanent memory is missing.

Ex. 1026, 172:10–22. When asked specifically how Perlman describes "permanent storage memory," Dr. Long responded, "I thought [Perlman] talked about flash in here. Certainly, it's something that a router can have and router[s] can and do have." Ex. 1026, 149:4–9. However, Dr. Long failed to identify how Perlman expressly discloses permanent storage memory. Based on his testimony, we conclude that a skilled artisan would understand Perlman to disclose routers that may or may not have permanent storage memory.

A finding of anticipation by inherency requires more than probabilities or possibilities. *Motorola Mobility LLC v. Int'l Trade Comm'n*, 737 F.3d 1345, 1350 (Fed. Cir. 2013). Based on the evidence discussed above, it is possible to infer that Perlman describes such permanent storage memory. However, Microsoft has not presented evidence that the computers or routers described by Perlman necessarily use permanent storage memory

as recited in claims 1 and 3. Therefore, Microsoft has failed to establish by a preponderance of evidence that Perlman describes a receiver/computer or sender/computer necessarily having "permanent storage memory." Without such evidence, we cannot conclude that Perlman anticipates claims 1 and 3.

(2) Claims 22–24

Proxyconn contends that Perlman fails to disclose the step recited in independent claim 22 of "searching for data with the same digital digest in said network cache memory." Resp. 20. Instead, Proxyconn urges that Perlman stores "one database identifier" that is merely "compared to the identifier received from the designated router." *Id.*; *see also* Ex. 2007, 13 (citing Ex. 1003, col. 8, ll. 32–42). Dr. Konchitsky testified, without citing any particular portion of Perlman, that Perlman's "receiving routers receive an identifier and each simply compares the received identifier with its existing identifier. The receiving routers are not searching for data files using the identifier as the key, or among multiple identifiers." Ex. 2002, ¶ 23.

Microsoft cites numerous passages from Perlman as meeting the recited "searching" step. '026 Petition, App'x. A, 17. All but one of the cited passages from Perlman describes the comparison of a received high-level database digest with a locally calculated high-level database identifier. The other cited passage (Ex. 1003, col. 8, ll. 22–49) describes an alternate embodiment in which received low-level digests are compared to locally generated low-level digests, if the received and locally generated high-level digests do not match. *See* part II.B.1 above (describing Perlman's use of high-level and low-level digests).

We conclude that Microsoft has failed to establish by a preponderance of evidence that Perlman meets the "searching" step recited in claim 22. Every portion of Perlman upon which Microsoft relies involves comparing one received digest to a corresponding locally calculated digest. These one-to-one comparisons cannot identify a particular data object from among a set of data objects as required by the "searching" step. Rather, the comparisons reveal whether a locally stored data object is synchronized with the corresponding remotely stored data object. Claims 23 and 24 depend from claim 22. We therefore, conclude that Perlman does not anticipate claims 22–24.

b. Yohe

(1) Claims 1, 3, and 10

Proxyconn first argues that Yohe fails to describe a sender/computer having permanent storage memory and means for creating digital digests of data. Resp. 21–23. Microsoft counters that the combination of Yohe's cache verifying computer 14 and file server 18 constitutes the claimed sender/computer. '026 Petition 16, App'x. A, 3; MS Reply 5–6; *see also* Ex. 1028, 16 (regarding annotated version of Yohe, Figure 2). We reject Proxyconn's argument because claims 1, 3, and 10 do not limit "sender/computer" to hardware residing in one housing. We have concluded that "sender/computer" may broadly encompass "multiple devices," and Proxyconn has directed us to no persuasive evidence that our interpretation is incorrect.

Proxyconn next argues that Yohe fails to describe a "receiver/computer" with means for creating digital digests on data. Resp. 23–24. Microsoft identifies Yohe's block signature generator 44 in

remote client computer 12 as the recited "means for creating digital digests on data." '026 Petition 20, App'x. A, 9; *see also* Ex. 1028, 13 (regarding annotated version of Yohe, Figure 2). We agree with Microsoft that Yohe's block signature generator 44 creates digests on data stored in Yohe's cache. Ex. 1005, col. 6, ll. 22–23.

Proxyconn also argues that Yohe fails to describe a "receiver/computer" with means for comparing received and locally generated digital digests on data. Resp. 24; *see also* Ex. 2007, 18–19 (regarding annotated version of Yohe, Figure 2). Proxyconn contends that comparator 58, which is part of cache verifying computer 14 and not remote client 12, compares the digests generated by block signature generators 44 and 56. Ex. 2007, 19 (annotating Yohe's Figure 2); *see also* Tr. 55:1–8 (identifying Ex. 1005, Yohe, col. 6, ll. 22 to col. 7, ll. 16 as confirming annotations on Yohe's Figure 2). Microsoft identifies Yohe's directory signature comparator 46 in remote client computer 12 as the recited "means for comparison between digital digests" that compares the digital digests on data that were separately created by the sender and receiver computers. '026 Petition 21, App'x. A, 10; *see also* Ex. 1028, 13 (regarding annotated version of Yohe, Figure 2).

We agree with Proxyconn that Yohe fails to describe a "receiver/computer including means for comparison of digital digests" as recited in claims 1 and 3. The two versions of digests on data that Yohe generates via block signature generators 44 and 56 are compared by comparator 58, which is not a component of remote client 12. For this reason, we conclude that Microsoft has not established by a preponderance of evidence that Yohe anticipates claims 1 and 3.

However, claim 10 does not require that the "means for comparison between digital digests" refers to comparing a digest generated by the receiver to a digest generated by the sender. *See* Part II.A.5.b above. Instead, we have concluded that the "means for comparison between digital digests" recited in claim 10 refers to structure that can compare *any* digital digest received from the network with another digital digest. Yohe's remote client 12 includes directory signature comparator 46 that compares a directory signature received from the cache verifying agent 54 (i.e., a portion of a sender/computer), with a directory signature retrieved from the remote client's cache. Ex. 1005, col. 8, ll. 5–11. Therefore, we agree with Microsoft that Yohe's directory signature comparator 46 is a "means for comparison between digital digests" as recited in claim 10. We determine that Microsoft has established by a preponderance of evidence that Yohe anticipates claim 10.

(2) Claims 6 and 7

Proxyconn argues that Yohe fails to describe a caching computer having network cache memory in its permanent storage memory as required in claims 6 and 7. Microsoft contends that permanent storage device 80, shown as part of Yohe's file server 18, constitutes the "permanent storage memory" of the claimed caching computer. '109 Petition 17, App'x. A,10. Essentially, Microsoft argues that the combination of Yohe's cache verifying computer 14 and file server 18 meet all the recited limitations for the caching computer of claim 6. Proxyconn contends that even if we were to agree that cache verifying computer 14, file server 18, and permanent storage device 80 were the claimed "caching computer," this combination still fails to describe network cache memory in the permanent storage

memory. Resp. 26–27. Microsoft fails to address this argument in its reply. *See* MS Reply 9. Our review of Yohe fails to identify any description of a network cache memory in permanent storage device 80. For this reason, we conclude that Microsoft has not established by a preponderance of evidence that Yohe anticipates claim 6 or its dependent claim 7.

(3) Claims 22 and 23

Proxyconn argues that Yohe fails to disclose the step recited in independent claim 22 of "searching for data with the same digital digest in said network cache memory." Resp. 27–28. Proxyconn contends that the only analysis of digests performed by Yohe's receiver (i.e., the remote client) is performed by Yohe's directory signature comparator 46, which compares one directory digest with a digest received from the cache verifying computer. *Id*. at 27. Proxyconn argues that such one-to-one comparisons are not "searching" as recited in claim 22. *Id*. at 27–28.

Microsoft does not dispute Proxyconn's characterization of the manner in which Yohe's directory signature comparator operates. MS Reply 11. Rather, Microsoft contends that Proxyconn's position is predicated on a flawed interpretation of "search" that requires more than a "single-comparison search." *Id*.

Based on our review of Yohe, the directory signature comparator 46 performs, at most, a series of one-to-one comparisons of received digests to locally generated digests for directory sub-objects. Ex. 1005, col. 7, l. 6 – col. 8, l. 25 (describing steps performed in DIRECTORY REQUEST function as shown in Figures 15 and 16). We conclude that such one-to-one comparisons do not identify a particular directory sub-object from among a set of directory sub-objects as required by the "searching" step. Rather, the

comparisons simply reveal whether a locally stored directory sub-object is synchronized with the corresponding remotely stored directory sub-object. Claim 23 depends from claim 22. We therefore conclude that Microsoft has not established by a preponderance of evidence that Yohe anticipates claims 22 and 23.

c. Santos

(1) Claims 1 and 3

Proxyconn raises two arguments allegedly distinguishing claims 1 and 3. We address each in turn below.

First, Proxyconn argues that Santos fails to describe a receiver/computer. Resp. 33–34. Proxyconn contends that Santos describes "a compressor and a decompressor, which are intermediate computers." *Id*. at 33. Microsoft counters that Proxyconn's argument rests upon an incorrect interpretation of "receiver/computer." We agree. We have interpreted "receiver/computer" to refer to "a computer that receives data." Santos's two computers on opposite ends of the communication channel can send and receive data. *See* Ex. 1004, 6 (stating that "[b]idirectional compression is achieved by using two instances of the protocol, one for each direction.") Thus, Santos uses "compressor" and "decompressor" to denote the function performed by each of two computers during transmission of specific data in a specific direction across the communication channel between them. Because data can move in both directions across that channel, both machines may function as a compressor (i.e., sender) or a decompressor (i.e., receiver). In the context of claims 1 and 3, the decompressor is acting as a "receiver/computer." We therefore reject Proxyconn's first argument.

Second, Proxyconn argues that Santos fails to describe "means for creating a digital digest on *data in the network cache memory*." Resp. 35. Proxyconn contends that because Santos's compressor and decompressor calculate the digest for data *after* the data is received but *before* the data is stored, the calculation of the digest is not on data in the network cache memory. *Id*. Proxyconn's argument rests upon an inferred interpretation of the creating means that requires a specific order of operations. More specifically, Proxyconn argues that "creating a digital digest on data in the network cache memory" implicitly requires that the receiver first read data from the network cache memory and then create a digest for that data.

Microsoft counters that claims 1 and 3 "do not require that the receiver first read the data from the cache and then and only then calculate a digest on it." MS Reply 8. Microsoft's argument also rests upon an inferred interpretation of the creating means. More specifically, Microsoft infers that the receiver's creating means need only be capable of creating a digest corresponding to data that is or will be stored in the network cache memory.

We conclude that Microsoft's interpretation is more consistent with the broadest reasonable interpretation of "means for creating digital digests on data in said network cache memory." Claim 1 and its dependent claim 3 recite systems, not methods, with hardware having recited functional capabilities. The functional language in claim 1 is not limited to a particular order of operations. "[A]pparatus claims cover what a device *is*, not what a device *does*. An invention need not *operate* differently than the prior art to be patentable, but need only *be* different." *Hewlett-Packard Co. v. Bausch & Lomb Inc.*, 909 F.2d 1464, 1468 (Fed. Cir. 1990); *see also Roberts v. Ryer*, 91 U.S. 150, 157 (1875) ("The inventor of a machine is entitled to the

benefit of all the uses to which it can be put, no matter whether he had conceived the idea of the use or not."). Here, the receiver/computer must be capable of creating a digest on data that is "in" the receiver's network cache memory.

We find that Santos describes a computer that can calculate a digest on data corresponding to data that will be stored in the receiver's network cache. When data not yet in Santos's compressor's cache is to be sent across the communication channel, Santos's compressor sends that data to the decompressor. Ex. 1004, 7. Santos's decompressor, upon receiving the data also creates a digest for that data and stores the digest and the data in the decompressor's cache. *Id*. Santos describes this case as follows:

> When the compressor receives a packet {HdrA, X} [i.e., data] to be forwarded over the link, where HdrA is the TCP/IP header and X is the data payload, it first computes H(X) [i.e., a digest], the fingerprint of X. If it finds that no entry indexed by H(X) [digest] exists in its cache, it stores X in its cache, indexed by H(X) [digest]. It then forwards the TCP/IP packet across the link. Upon receiving a TCP/IP packet forwarded over the channel, the decompressor also computes H(X) [digest], and stores X in its cache, indexed by H(X) [digest]. The TCP/IP packet is then output from the system.

Ex. 1004, 7, Figure 4.

Santos's decompressor thus calculates a digest for every data payload that it receives from the compressor and stores that data payload and its digest in its network cache memory. Ex. 1004, 7–8. We therefore reject Proxyconn's second argument and find that Santos describes a receiver/computer that includes "means for creating digital digests on data in said network cache memory." We determine that Microsoft has established by a preponderance of evidence that Santos anticipates claims 1 and 3.

(2) Claim 10

Proxyconn contends that Microsoft never challenged the patentability of claim 10 as anticipated by Santos and that our decision to institute a trial on this ground was "a mistake." Resp. 32–33, n.5. For this reason, Proxyconn never substantively addresses Santos as it relates to claim 10. *Id*. at 32–33. However, Microsoft's claim charts compared claim 10 to Santos. '026 Petition, App'x. A 11–13. That claim chart is part of Microsoft's petition, and we instituted review of claim 10 as anticipated by Santos on the grounds raised in the claim chart. Microsoft has thus proffered evidence to establish that Santos describes every limitation of claim 10. Proxyconn fails to rebut Microsoft's evidence. In the absence of countervailing evidence and argument, we determine that Microsoft has established by a preponderance of evidence that Santos anticipates claim 10.

(3) Claim 22

Claim 22 is directed to a method "performed by a receiver/computer." Ex. 1002, col. 12, ll. 30–31. Just as with claims 1 and 3, Proxyconn argues that Santos fails to describe a "receiver/computer." Resp. 33–34. For the same reasons described above in connection with claims 1 and 3, we find that Santos describes the claimed receiver/computer. Microsoft has proffered evidence to establish that Santos meets every other limitation of claim 22, and Proxyconn fails otherwise to rebut that evidence. Therefore, we determine that Microsoft has established by a preponderance of evidence that Santos anticipates claim 22.

(4) Claim 23

Proxyconn argues that Santos fails to describe "searching in predetermined locations of said permanent storage memory for data" for two reasons. Resp. 35–36. First, Proxyconn argues that Santos "lacks permanent storage memory," which is evident from the loss of "fingerprints H(X) and associated data" when Santos's computers are reset. *Id*. at 35 (citing "Ex. 1004 at § 3.2.1").[4] Proxyconn contends "[i]f either the data payload or fingerprints H(x) were stored in permanent storage memory, the data and fingerprint H(x) would be retained in memory, following either a reset or power cycle." Resp. 35–36.

In response, Microsoft characterizes Santos's compressor and decompressor as "general-purpose" computers that "necessarily" have ROM and a hard disk. MS Reply 11. Microsoft also contends that Santos' description of a 200 MB cache in a system with only 128 MB RAM implies that some cache must be in non-volatile memory. *Id*. (citing Ex. 1004, 7–9 and Figures 4 and 5). Microsoft does not directly address Proxyconn's contention that the loss of data in Santos's cache upon a reset demonstrates that Santos's cache does not reside in permanent storage memory.

Based on our own review of Santos, we reject Proxyconn's argument as not being supported by the evidence. Santos's only description of a reset relates to a "reset message" rather than a system reset or power interruption. Ex. 1004, 7 and 9. Santos describes using a reset message sent from the compressor to the decompressor or vice versa as a mechanism for handling a

[4] The cited portion of Santos does not address the effect of a "reset." Nonetheless, Santos describes a reset process elsewhere. Ex. 1004, 9 (in § 3.3).

lack of synchronization between the caches in each machine. *Id*. at 9. We understand these reset messages to be sent intentionally to reset the other cache to a known state in the event that the stored information about "illegal fingerprints" is lost (e.g., due to a compressor restart). *Id.* Santos does not describe where or how the compressor stores information relating to illegal fingerprints. Thus, we find Proxyconn's cited evidence to be inconclusive regarding the character of the cache memory.

Microsoft's evidence on the nature of the cache memory is more persuasive. Santos describes its compressor and decompressor in some detail as being "Intel-based Pentium II 300 MHz machines running Linux 2.0.31 with 128 MB of RAM." Ex. 1004, 9. Santos also states: "we limited the amount of memory available for the caches, . . . , to 200 MB each." *Id*. Santos also expressly describes repeated and numerous write operations to the cache. Ex. 1004, Figures 4 and 5 and accompanying text at 7–8.

We are persuaded that a skilled artisan would understand that Santos's 200 MB cache, which exceeds the available RAM in each machine, referred to a non-volatile memory that supports multiple write operations, which satisfies our interpretation of "permanent storage memory." We therefore reject Proxyconn's argument that Santos does not describe "permanent storage memory."

Second, Proxyconn argues that Santos fails to "search in predetermined locations" within the permanent storage memory. Resp. 36. Proxyconn asserts that "[s]earching in an index or files stored in memory is far different than searching at a predetermined location in memory, much less in predetermined locations in permanent storage memory." *Id*. The sole

support that Proxyconn provides for this assertion is the following testimony of Dr. Konchitsky.

> 51. I understand Santos to disclose that the fingerprints are not stored in permanent storage memory. Santos states that upon reset, such as through a power cycle or restart, the fingerprints H(x) and associated data is lost. [EX1004] at §3.3. This indicates to me, as it would to any person of ordinary skill in the art[,] that the fingerprints are not stored in permanent storage memory, and thus Santos could not logically teach searching permanent storage memory for fingerprints.

Ex. 2002 ¶ 51, Resp. 36. Dr. Konchitsky's testimony does not support Proxyconn's assertion that Santos searches "in predetermined locations" in memory. Rather, Dr. Konchitshy's testimony relates to whether Santos describes "permanent storage memory."

Microsoft responds that Santos "looks in at least two predetermined locations: it looks at H(X) and then, to identify any collisions, it looks at the stored payload associated with the H(X) and compares that to the argument payload." MS Reply 11 (citing Ex. 1004, 7–9, Figures 4 and 5). We determine that the evidence cited by Microsoft is persuasive and, therefore, reject Proxyconn's argument that Santos does not perform "searching in predetermined locations."

For the foregoing reasons, we conclude that Microsoft has established by a preponderance of evidence that Santos anticipates claim 23.

d. DRP

(1) Claims 6, 7, and 9

Proxyconn argues that DRP does not anticipate claims 6, 7, and 9 because it fails to describe the following three limitations on the caching computer recited in claim 6: (i) a "permanent storage memory;" (ii) "means

for comparison;" and (iii) "means for calculating a digital digest" as recited in claim 6. Resp. 37–38. Proxyconn argues that DRP also fails to describe additional limitations on the caching computer that are recited in dependent claims 7 and 9.[5] However, those arguments rely upon Proxyconn's contention that DRP fails to disclose a network cache in permanent storage memory. Resp. 38–39. Microsoft contends that DRP describes the claimed caching computer as any of its client and server computers, all of which have "permanent storage memory," "means for comparison," and "means for calculating a digital digest." MS Reply 8–9. Microsoft also contends that DRP's client and server computers meet each of the additional limitations on the caching computer that are recited in dependent claims 7 and 9.

Proxyconn's argument rests upon Proxyconn's interpretation of "two other computers" as excluding the "caching computer." We have rejected Proxyconn's interpretation as explained in Part II.A.4 above. Proxyconn does not otherwise dispute Microsoft's characterization of DRP's client or server computers as meeting all the limitations on the "caching computer" that are recited in claims 6, 7, and 9. Therefore, we determine that Microsoft has established by a preponderance of evidence that DRP anticipates claims 6, 7, and 9.

---

[5] Claim 7 depends upon claim 6 and further recites "said caching computer further includes means for calculating a digital digest for data in its network cache memory." Ex. 1002, col. 11, ll. 13–15. Claim 9 also depends upon claim 6 and further recites "said caching computer further includes means for storing said digital digest in said permanent storage memory." *Id.* at col. 11, ll. 18–20.

(2) Claims 11, 12, and 14

Microsoft identifies in great detail how DRP's clients and servers communicate according to the methods of claims 11, 12, and 14. '109 Petition, Appendix A 16–22; MS Reply 9–10 (citing DRP. 4:37–5:19, 5:22–32, 6:40–7:1, 7:20–29, 7:31–35, 7:37–39, 8:11–13, 8:29–31, 9:22–32; Konchitsky Tr. 91:18–94:7, 98:5–11, 108:11–109:18). Proxyconn argues that DRP fails to describe the step of "receiving a response signal from said receiver/computer at said sender/computer, said response signal containing a positive, partial or negative indication signal for said digital digest, and if a negative indication signal is received, transmitting said data from said sender/computer to said receiver/computer" as recited in independent claim 11. Resp. 40. Proxyconn contends that DRP's statement that the "client can use the index to automatically download the files that are specified" means that the client downloads specific files without ever sending a response signal to the server. *Id.* Proxyconn also argues that DRP fails to describe the requirement in claim 14 of "a response signal is sent containing a separate indication signal for each of said data objects." *Id.* at 41.

We reject Proxyconn's arguments regarding both claims. DRP describes that the client, after comparing the received digest of files with the digest for its cached versions, sends a GET request (when none of the digests for files match), a differential GET request (when some, but not all, of the digests for files in the cache match), or no request (when all digests for the files in the cache match). DRP, 5:22–32, 6:44, 7:20–28, 8:11–13. Dr. Konchitsky confirms the accuracy of Microsoft's position regarding the GET and differential GET requests that the client might send to the server. Konchitsky Tr. 108:11–109:18. The three types of responses that the client

sends to the server after receiving the index from the server and comparing it to the local index constitute the claimed "response signal containing a positive, partial or negative indication signal" of claim 11. These types of responses also correlate to the "separate indication signal" of claim 14. Proxyconn proffers no argument independently distinguishing claim 12 from DRP. Therefore, we conclude that DRP anticipates claims 11, 12, and 14.

2. Obviousness

"Section 103 [of 35 U.S.C.] forbids issuance of a patent when 'the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.'" *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). To establish obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *See CFMT, Inc. v. Yieldup Int'l Corp.*, 349 F.3d 1333, 1342 (Fed. Cir. 2003); *In re Royka*, 490 F.2d 981, 985 (CCPA 1974).

a. Yohe and Perlman

Microsoft contends that the combination of Yohe and Perlman renders claims 1, 3, 10, and 22–24 obvious under 35 U.S.C. § 103(a). Dr. Long testified that a skilled artisan would have considered Yohe and Perlman to be closely related technologies that are natural to combine because both references address the same problem and use the same algorithm in similar applications. Ex. 1007, 9:12–10:13. Dr. Long also testified that Perlman expressly suggests that its technology is suited for use in any type of node in a network for which synchronization of data is important. *Id.* at 11:1–7.

Proxyconn does not address the sufficiency of the combined teachings of Yohe and Perlman regarding the subject matter of claims 1, 3, 10, and 22–24. Instead, Proxyconn first argues that Perlman is not a proper reference to consider in an obviousness analysis because a skilled artisan would not consider Perlman to be analogous art to the claimed invention. Resp. 28–31. Dr. Konchitsky testified that the ’717 Patent addresses increasing data access speed by conserving bandwidth while Perlman uses additional bandwidth to keep its nodes synchronized. Ex. 2002, 6. Dr. Konchitsky concluded that a skilled artisan would not consider Perlman to describe a viable way of increasing data access.

Dr. Long testified that Perlman, Yohe, and the ’717 Patent all address the same problem: “the desire to reduce redundancy in network data transmissions where dynamic data previously sent over the network has been stored by the receiver for possible later reuse.” Ex. 1007, 9:13–15. We find Dr. Long’s statement of the problem addressed by the ’717 Patent, Perlman, and Yohe to be persuasive. The ’717 Patent states: “The performance gains realized by the present invention are derived from the fact that computers in common wide-area networks tend to repetitively transmit the same data over the network.” Ex. 1002, col. 6, ll. 17–20. Both Perlman and Yohe describe reducing the use of bandwidth for data transmission as a way of improving network performance. Ex. 1007, 9:19–10:4 (citing Ex. 1003, col. 3, ll. 52–55; Ex. 1005, col. 4, ll. 32–40). Therefore, we reject Proxyconn’s argument that a skilled artisan would not consider Perlman’s teachings to be pertinent to the invention.

Proxyconn also argues that it would not have been obvious to incorporate Yohe’s permanent storage memory into Perlman’s router

because such memory would serve no function in Perlman's router. Resp. 31–32. "The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference. . . . Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art." *In re Keller*, 642 F.2d 413, 425 (CCPA 1981) (citations omitted). Dr. Long testified at length about the reasons why a skilled artisan would have used permanent storage memory as taught by Yohe for the cache in Perlman. Ex. 1007, 12:4–15:9. Proxyconn proffers no persuasive evidence in support of its position. Therefore, we reject Proxyconn's argument that a skilled artisan would not have found it obvious to use permanent storage memory in the receiver/computer and sender/computer.

For the combination of Yohe and Perlman to render claims 1, 3, 10, and 22–24 obvious, the combination still must describe or suggest all limitations of the claims. Therefore, we analyze the teachings and the differences, if any, between the combination of Yohe and Perlman and the claims below.

(1) Claims 1, 3, and 10

As discussed in Part II.C.1 above, we determine that Yohe anticipates claim 10, but that neither Yohe nor Perlman anticipates claims 1 and 3. The only limitation recited in claims 1 and 3 not described by Yohe is the "means for comparison of digital digests on data" located in the receiver/computer. *See* Part II.C.1.b(1) above. However, Yohe describes comparator 58 in cache verifying computer 14 that would, if located in the remote client, meet the recited "means for comparison of digital digests." Additionally, Perlman describes performing the comparison of digital

digests on data in the receiver because Perlman's receiving router compares a received digest with a locally generated and stored digest.
Ex. 1003, col. 7, ll. 56–63.

Thus, we must decide whether a skilled artisan would have found it obvious to incorporate Perlman's comparing means into Yohe's remote client or to move or add Yohe's comparator 58 from cache verifying computer 14 into remote client 12. Both Perlman and Yohe describe devices that compare digital digests on data. Ex. 1003, col. 7, ll. 56–63; Ex. 1005, col. 6, ll. 22–37, Figures 6 and 7. Yohe suggests that "[i]t is recognized that other locations for the comparator [58] may exist." Ex. 1005, col. 13, ll. 32–34. Proxyconn has not suggested that Yohe's comparator would not be capable of performing the recited function of "comparison of digital digests on data." Incorporating Perlman's comparing means into Yohe's remote client or moving Yohe's comparator 58 into the remote client would have involved nothing more than ordinary skill and would have been using known devices to perform known functions to yield predictable results. We conclude that a skilled artisan would have found it obvious to include the recited "means for comparison of digital digests on data" in a receiver/computer. For these reasons, we conclude that Microsoft has established by a preponderance of evidence that the combination of Yohe and Perlman renders claims 1, 3, and 10 unpatentable as obvious under 35 U.S.C. § 103(a).

(2) Claims 22–24

As explained in Parts II.C.1.a(2) and II.C.1.b(3) above, neither Perlman nor Yohe describes, "searching" as required by claim 22. We cannot conclude that a claim would have been obvious when the prior art

does not describe every element recited in the claim. Claims 23 and 24 depend from claim 22. Therefore, we conclude that Microsoft has not established by a preponderance of evidence that claims 22–24 are unpatentable as obvious under 35 U.S.C. § 103(a).

b. Mattis and DRP

For the reasons expressed in Part II.C.1.d above, we determine that DRP anticipates claims 6, 7, 9, 11, 12, and 14. Because we cancel these claims based on DRP alone, we determine that Microsoft's challenge to the patentability of claims 6, 7, 9, 11, 12, and 14 as being obvious in light of Mattis combined with DRP is moot and do not address this challenge.

D. Proxyconn's Motion to Amend

Proxyconn moved to substitute claims 35–41[6] for challenged claims 1, 3, 6, 10, 11, 22, and 23, respectively, if the Board were to cancel any of those challenged claims as unpatentable. Mot. Amend 1. Proxyconn also requests that we enter claims 35–41 "in addition to the original claims." *Id*. Proxyconn may not add a proposed claim while retaining the original claim for which the proposed claim is substituted. 35 U.S.C. § 316(d)(1)(B) and 37 C.F.R. § 42.121(a)(3). Therefore, to the extent that Proxyconn's Motion to Amend requests entry of substitute claims in addition to the original claims, we deny the Motion.

[6] Proxyconn mistakenly refers to substitute claims 35–42 in its Motion to Amend, but proffers only claims 35–41 as substitutes for claims 1, 3, 6, 10, 11, 22, and 23, respectively. Mot. Amend, Appendix A.

1. *Proxyconn's Burden of Persuasion Relating to Claims 35–41*

Inter partes review is neither examination nor reexamination of a patent application. We do not enter proposed substitute claims as a matter of right. Rather, the patent owner must prove its entitlement to the proposed claims. Rule 42.20(c) states: "(c) *Burden of proof.* The moving party has the burden of proof to establish that it is entitled to the requested relief." 37 C.F.R. § 42.20(c). We set forth the requirements for demonstrating the prima facie patentability of substitute claims in *Idle Free Sys., Inc. v. Bergstrom, Inc.*, IPR2012-00027, Paper 26, as follows:

> A patent owner should identify specifically the feature or features added to each substitute claim, as compared to the challenged claim it replaces, and come forward with technical facts and reasoning about those feature(s), including construction of new claim terms, sufficient to persuade the Board that the proposed substitute claim is patentable over the prior art of record, and over prior art not of record but known to the patent owner. The burden is not on the petitioner to show unpatentability, but on the patent owner to show patentable distinction over the prior art of record and also prior art known to the patent owner. Some representation should be made about the specific technical disclosure of the closest prior art known to the patent owner, and not just a conclusory remark that no prior art known to the patent owner renders obvious the proposed substitute claims.
>
> A showing of patentable distinction can rely on declaration testimony of a technical expert about the significance and usefulness of the feature(s) added by the proposed substitute claim, from the perspective of one with ordinary skill in the art, and also on the level of ordinary skill, in terms of ordinary creativity and the basic skill set. A mere conclusory statement by counsel, in the motion to amend, to the effect that one or more added features are not described in any

> prior art, and would not have been suggested or rendered obvious by prior art, is on its face inadequate.

Idle Free Sys., Inc. v. Bergstrom, Inc., IPR2012-00027, slip op. at 7–8 (PTAB June 11, 2013 (Paper 26)); *see also Idle Free Sys., Inc. v. Bergstrom, Inc.*, IPR2012-00027, slip op. 33–38 (PTAB January 7, 2014 (Paper 66)).

Proxyconn has not proffered sufficient arguments or evidence to establish a prima facie case for the patentability of claims 35–41. For example, Proxyconn has not: (i) construed the newly added claim terms; (ii) addressed the manner in which the claims are patentable generally over the art; (iii) identified the closest prior art known to it; (iv) addressed the level of ordinary skill in the art at the time of the invention; or (v) discussed how such a skilled artisan would have viewed the newly recited elements in claims 35–41 in light of what was known in the art. Instead, Proxyconn attempts to distinguish claims 35–41 only from the prior art for which we instituted review of corresponding claims 1, 3, 6, 10, 11, 22, and 23. Mot. Amend 4–15. Consequently, Proxyconn has failed to establish a prima facie case for the patentability of claims 35–41. Proxyconn's motion to amend in connection with claims 35–41 is, therefore, denied on these grounds. We also find Proxyconn's motion unavailing for additional reasons raised by Microsoft as described below.

2. *Patentability of Claims 35–41 in light of DRP*

a. Claims 35, 36, 38, 40, and 41

Microsoft contends that DRP anticipates claims 35, 36, 38, and 40–41 and supports its contentions with detailed citations to DRP. Microsoft Corporation's Opposition to Patent Owner's Corrected Motion to Amend under 37 C.F.R. § 42.121 (Paper 48) 1–15 ("MS Amend Opp."). In its

Motion to Amend, Proxyconn does not compare claims 35, 36, 38, 40, and 41 to DRP. Mot. Amend 4–13. Proxyconn argues that Microsoft improperly injects a "new ground of review" into the trial by asserting that DRP would anticipate claims 35, 36, 38, 40, and 41 because these claims are amended versions of claims 1, 3, 10, 22, and 23 respectively, for which no challenge based on DRP has been instituted. Mot. Amend Reply 2–3.

Proxyconn's argument is unpersuasive. "Petitioners may respond to new issues arising from proposed substitute claims including evidence responsive to the amendment. 35 U.S.C. 316(a) and 326(a). This includes the submission of new expert declarations that are directed to the proposed substitute claims." Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,767 (August 14, 2012).[7] Microsoft has provided evidence—i.e., specific citations to portions of DRP—that responds to new issues introduced by Proxyconn's proposed substitute claims. Microsoft is entitled to do so. Proxyconn provides no evidence to counter Microsoft's contentions that DRP anticipates claims 35, 36, 38, 40, and 41. Proxyconn carries the burden of proof with respect to the patentability of its proposed claims. 37 C.F.R. § 42.20(c). Because the only evidence of record supports Microsoft's position, we conclude that Proxyconn has failed to establish by a preponderance of evidence that claims 35, 36, 38, 40, and 41 are patentable

[7] Proxyconn's Motion to Amend was filed after publication of our decision in *Idle Free Sys., Inc. v. Bergstrom, Inc.*, IPR2012-00027, Paper 26, discussed above, which set forth the requirements for meeting the burden of proof on a motion to amend. We also reminded Proxyconn at the oral hearing of its duty to distinguish the proposed claims from all prior art of which it is aware, including DRP. Tr. 64:3–13.

over DRP. Therefore, we deny the Motion to Amend as it relates to claims 35, 36, 38, 40, and 41.

b. Claim 37

Claim 37, which is substituted for claim 6, is reproduced below with additions indicated by underlining.

> 37. A system for data access in a packet-switched network, comprising:
>
> a gateway including an operating unit, a memory and a processor connected to said packet-switched network in such a way that network packets sent between at least two other computers pass through it;
>
> a caching computer connected to said gateway through a fast local network, wherein said caching computer includes an operating unit, a first memory, a permanent storage memory and a processor; said caching computer further including a network cache memory in its permanent storage memory, means for calculating a digital digest <u>on data</u> and means for comparison between a digital digest on data in its network cache memory and a digital digest received from said packet-switched network through said gateway<u>, wherein said data includes a plurality of octet ranges in a file or files</u>.

Mot. Amend, App. A, 1–2.

We have determined that DRP anticipates claim 6. *See* Part II.C.1.d(1) above. Proxyconn's entire argument for distinguishing claim 37 from DRP is: "Substitute claim 37 requires structure to operate a data including a plurality of octet ranges in a file or files. DRP discloses content identifier based on information objects . . . . Lacking this additional element, DRP . . . fail[s] to anticipate and/or render obvious Substitute Claim 37." Mot. Amend 8–9. Proxyconn fails to cite any support for its characterization of DRP. Microsoft responds: "The data processed in DRP

includes plural files and thus includes a plurality of ranges of octets in plural files. . . . The 'data that is distributed . . . can consist of any kind of code or content.'" MS Amend Opp. 6 (citing DRP, 2:31–32, 2:44–3:2, 3:14–16, 3:28–31) (internal citation omitted). Because we agree with Microsoft, we determine that Proxyconn has failed to establish by a preponderance of evidence that claim 37 is patentable over DRP.

c. Claim 39

Claim 39, which is substituted for claim 11, is reproduced below with additions indicated by underlining and deletions indicated by double square bracketing.

> 39. A method performed by a sender/computer in a packet-switched network for increasing data access, said sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and said sender/computer being operative to transmit data to a receiver/computer, the method comprising the steps of:
>
> creating a digital digest on data; [[and]]
>
> receiving a request for said data from the receiver/computer;
>
> in response to the request for data, transmitting said [[a]] digital digest of said data from said sender/computer to said receiver/computer;
>
> receiving a response signal from said receiver/computer at said sender/computer, said response signal containing a positive, partial or negative indication signal for said digital digest, and
>
> if a negative indication signal is received, transmitting said data from said sender/computer to said receiver/computer.

Mot. Amend, App., A 2–3.

Proxyconn contends, without citing any portion of DRP in support,[8] that DRP fails to describe "the step of receiving a request for data, and in response to the request for data, transmitting a digital digest for the data." Mot. Amend 10. In response, Microsoft contends that DRP describes this missing step being performed by the server responding to a GET request from the client for an index file with the current version of the index. MS Amend Opp. 10–11 (citing DRP, 5:22–32, 6:43–7:1, 7:20–31, 7:37–38, 8:8–13, 9:22–32). We determine that on the evidence before us, Proxyconn has failed to establish by a preponderance of evidence that claim 39 is patentable over DRP.

3. Alleged Broadening of Scope in Claims 36, 38, 40, and 41

During *inter partes* review, a patent owner may not amend a challenged claim in a manner that enlarges the scope of that claim. 35 U.S.C. § 316(d)(3); 37 C.F.R. § 42.121(a)(2)(ii). Proxyconn states without further discussion that "the amendments herein do not seek to enlarge the scope of the claims of the '717 Patent." Mot. Amend 1. Microsoft argues that Proxyconn's proposed claims 36, 38, 40, and 41 impermissibly enlarge the scope of challenged claims 3, 10, 22, and 23 respectively. MS Amend Opp. 4, 7. We address each of these claims in turn below.

[8] Proxyconn cites section II.F (sic, II.G) of Proxyconn's Patent Owner Response. Mot. Amend 10. However, this section of the Response cites only ¶¶ 52–61 of Dr. Konchitsky's Declaration. Resp. 36–41. None of those paragraphs cites any portion of DRP to support Dr. Konchitsky's testimony. *See* Ex. 2002, Konchitsky Decl. ¶¶ 52–61.

a. Claim 36

Claim 36, which is substituted for claim 3, is reproduced below with additions indicated by underlining and deletions indicated using strikethrough or double square bracketing.

> 36. A system for data access in a packet-switched network, comprising[[:]] a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor; and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor[[,]]; said sender/computer and said receiver/computer ~~communicating through said network;~~ configured to communicate with one another through said network; said sender/computer ~~further~~ includ~~es~~ing means for creating digital digests on data[[;]] stored in said permanent storage memory, and said receiver/computer ~~further~~ including a network cache memory ~~and~~, means for creating digital digests on data in said network cache memory[[;]], and ~~said receiver/computer including~~ means for ~~comparison~~ comparing between digital digests created by the sender/computer and receiver/computer; wherein said receiver/computer further includes means for storing ~~said created~~ at least one of the digital digests created by the sender/computer in its ~~first or~~ permanent storage memory; wherein the data includes at least a range of octets in a file.

See Mot. Amend, App. A, 1.

Microsoft argues that Proxyconn enlarges claim 36 in two ways. First, the claim is allegedly enlarged by changing the requirement that the sender/computer and receiver/computer are "communicating through said network" to two computers that are "configured to communicate with one another through said network." MS Amend Opp. 4. We agree that claim 36 no longer requires that the sender and receiver computers are communicating through the network, which would necessarily also require

that the computers be "configured" to so communicate. Claim 36 more broadly covers sender and receiver computers that are configured to communicate, but are not necessarily communicating. Doing so impermissibly enlarges the scope of claim 3. Therefore, we deny Proxyconn's motion to amend regarding claim 36, and need not reach Microsoft's second argument.

b. Claim 38

Claim 38, which is substituted for claim 10, is reproduced below with additions indicated by underlining and deletions indicated using strikethrough or double square bracketing.

> 38. A system for data access in a packet-switched network, comprising:
>
> a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, said sender/computer and said receiver/computer communicating through a network;
>
> said sender/computer further including means for creating digital digests on data, and
>
> said receiver/computer further including a network cache memory, means for storing [[a]] at least one of said digital digest received from said network in its permanent storage memory, and said receiver/computer configured to search for a digital digest received from the sender/computer, in response to receiving the digital digest, ~~means for comparison between digital digests;~~ wherein said data includes at least a range of octets in a file.

Mot. Amend, App. A, 2.

Microsoft argues that "said receiver/computer configured to search for a digital digest received from the sender/computer, in response to receiving the digital digest" covers subject matter not covered by the receiver/computer's "means for comparison between digital digests" recited in claim 10. MS Amend Opp. 6–7. Microsoft contends, "[f]or example, the receiver might check for a given digest by comparing hashes (or other identifiers) of digital digests not the digital digests themselves." *Id.* at 7. We agree. Claim 38 covers a receiver/computer that can search for a digital digest in ways other than comparing the digital digests themselves. Therefore, we deny Proxyconn's motion to amend regarding claim 38.

c. Claims 40 and 41

Claim 40, which is substituted for claim 22, is reproduced below with additions indicated by underlining and deletions indicated using strikethrough or double square bracketing.

> 40. A method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of:
>
> <u>sending a request for data</u>;
>
> receiving a message containing a digital digest for the requested data from said network;
>
> searching for<u> each received digital digest </u>~~data with the same digital digest~~ in said network cache memory,
>
> if ~~data having~~ the same digital digest as the digital digest received is not uncovered, forming a negative indication signal and transmitting <u>the negative indication signal</u> [[it]] back through said network; and

> creating a digital digest for data received from the sender/computer and stored in said network cache memory.

Mot. Amend, Appendix A 3.

Claim 41, which is substituted for claim 23, is reproduced below with additions indicated by underlining and deletions indicated using strikethrough.

> 41. The method as claimed in ~~claim 22,~~ claim 40, wherein searching in said network cache memory includes ~~further comprising~~ searching in predetermined locations in said permanent storage memory for data with a digital digest ~~substantially~~ identical to the searched one of the digital digests received from said network.

Id.

Microsoft argues that the third and fourth changes to claim 22 reflected in proposed claim 40 impermissibly enlarge the scope of claim 22. Microsoft contends that "[c]laim 40 newly covers methods that look only for matching digests but not for data having those digests. Claim 22 does not cover such a method." MS Amend Opp. 11. Proxyconn responds that "searching specifically for 'each received digital digest,' . . . is narrower than merely searching generally for data by the digital digest." Mot. Amend Reply 5. Proxyconn's argument mischaracterizes claim 22 as requiring "searching generally for data by the digital digest." Claim 22 recites: "searching for data with the same digital digest" not "by the same digest."

The question presented by the parties' arguments is whether it is possible to search for each received digital digest without searching for data having the same digital digest. Microsoft does not identify an example of how one might search for data with the same digest without using the digest. However, Proxyconn does not provide evidence that it is not possible to

search for each received digest without searching for data having the same digest. Without such evidence, we can only evaluate the scope of the claim based on the plain meaning of the terms. On that basis, we conclude that it would be possible to search for each received digital digest without searching for data having the same digital digest. Therefore, it is possible to practice the method recited in claim 40 without practicing the method recited in claim 22. For this reason, we conclude that claim 40 is impermissibly broader than claim 22. The same flaw exists in claim 41, which depends from claim 40. Therefore, we deny Proxyconn's Motion to Amend as it relates to claims 40 and 41.

III. CONCLUSION

Microsoft has established by a preponderance of evidence that claims 1, 3, 6, 7, 9, 10, 11, 12, 14, 22, and 23 are unpatentable as anticipated and claims 1, 3, and 10 are unpatentable as being directed to obvious subject matter. Microsoft has not established by a preponderance of evidence that claim 24 is unpatentable.

IV. ORDER

It is ORDERED that:

Claims 1, 3, 6, 7, 9–12, 14, 22, and 23 are CANCELED; and

Proxyconn's Motion to Amend Claims is DENIED.

PATENT OWNER:

Matthew L. Cutler
Bryan K. Wheelock
Douglas A. Robinson
HARNESS, DICKEY & PIERCE, PLC
mcutler@hdp.com
bwheelock@hdp.com
drobinson@hdp.com

PETITIONER:

John D. Vandenberg
Stephen J. Joncus
KLARQUIST SPARKMAN LLP
john.vandenberg@klarquist.com
stephen.joncus@klarquist.com

US006757717B1

(12) **United States Patent**
Goldstein

(10) **Patent No.: US 6,757,717 B1**
(45) **Date of Patent: Jun. 29, 2004**

(54) **SYSTEM AND METHOD FOR DATA ACCESS**

(75) Inventor: **Leonid Goldstein**, Herzlia (IL)

(73) Assignee: **ProxyConn, Inc.**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/398,007**

(22) Filed: **Sep. 16, 1999**

(51) **Int. Cl.**[7] **G06F 15/16**
(52) **U.S. Cl.** **709/217**; 707/1; 707/10; 709/203; 709/225; 709/229
(58) **Field of Search** 709/217, 203, 709/229, 225; 707/10, 1

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,897,781 A 1/1990 Chang et al.
5,384,565 A 1/1995 Gregory
5,682,514 A * 10/1997 Yohe et al. 707/1
5,732,265 A 3/1998 Dewitt et al.
5,835,943 A 11/1998 Yohe et al.
5,852,717 A 12/1998 Bhide et al.
5,864,837 A 1/1999 Maimone
5,878,213 A * 3/1999 Bittinger et al. 707/10
5,909,569 A * 6/1999 Housel et al. 707/10
5,919,247 A * 7/1999 Van Hoff et al. 709/217
5,924,116 A 7/1999 Aggarwal et al.
5,931,947 A * 8/1999 Burns et al. 713/201
5,978,791 A 11/1999 Farber et al.
6,003,087 A * 12/1999 Housel et al. 707/203
6,006,034 A 12/1999 Heath et al.
6,012,085 A 1/2000 Yohe et al.
6,021,491 A * 2/2000 Renaud 713/179
6,073,173 A * 6/2000 Bittinger et al. 707/513
6,085,249 A * 7/2000 Wang et al. 709/229
6,101,543 A * 8/2000 Alden et al. 709/229
6,122,637 A * 9/2000 Yohe et al. 709/204
6,134,583 A * 10/2000 Herriot 709/217
6,148,340 A * 11/2000 Bittinger et al. 709/203
6,256,664 B1 * 7/2001 Donoho et al. 709/204
6,279,041 B1 * 8/2001 Baber et al. 709/232

OTHER PUBLICATIONS

RFC2068: Hypertext Transfer Protocol—HTTP/1.1, Jan. 1997, pp. 9–12, 29–35, 70–94, 101–138.
RFC791: Internet Protocol Darpa Internet Program Protocol Specification, Sep. 1981, pp. 3, 11–14, 26.
RFC761: DOD Standard Transmission Control Protocol, Jan. 1980, pp.4, 15–17.
RFC1826: IP Authentication Header, Aug. 1995, pp. 1–10.
RFC1864: The Content–MD5 Header Field, Oct. 1995, pp.1–2.
RFC1321: The MD5 Message–Digest Algorithm, Apr. 1992, pp. 1–21.
Dave Fogle, Ethernet Frame Types, article from Feb. 1996 issue of LAN Magazine/Network Magazine, pp. 1–4.
W. Richard Stevens, TCP/IP Illustrated, vol. 1, 1994, pp. 33–37, 143–147, 223–228.

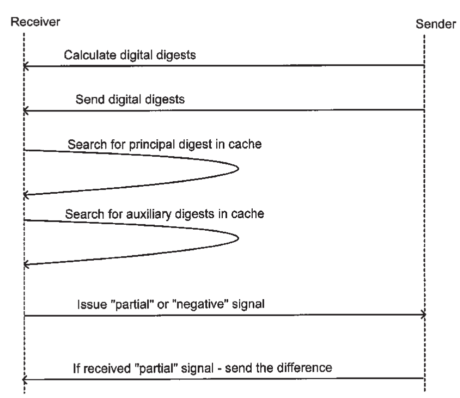
* cited by examiner

Primary Examiner—Jack B. Harvey
Assistant Examiner—Hai V. Nguyen
(74) *Attorney, Agent, or Firm*—J. D. Harriman, II, Esq.; Coudert Brothers LLP

(57) **ABSTRACT**

The invention provides a system for data access in a packet-switched network, including a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, the sender/computer and the receiver/computer communicating through the network; the sender/computer further including device for calculating digital digests on data; the receiver/computer further including a network cache memory and device for calculating digital digests on data in the network cache memory; and the receiver/computer and/or the sender/computer including device for comparison between digital digests. The invention also provides a method and apparatus for increased data access in a packet-switched network.

34 Claims, 9 Drawing Sheets



MICROSOFT
EXHIBIT 1002

Page 1 of 19

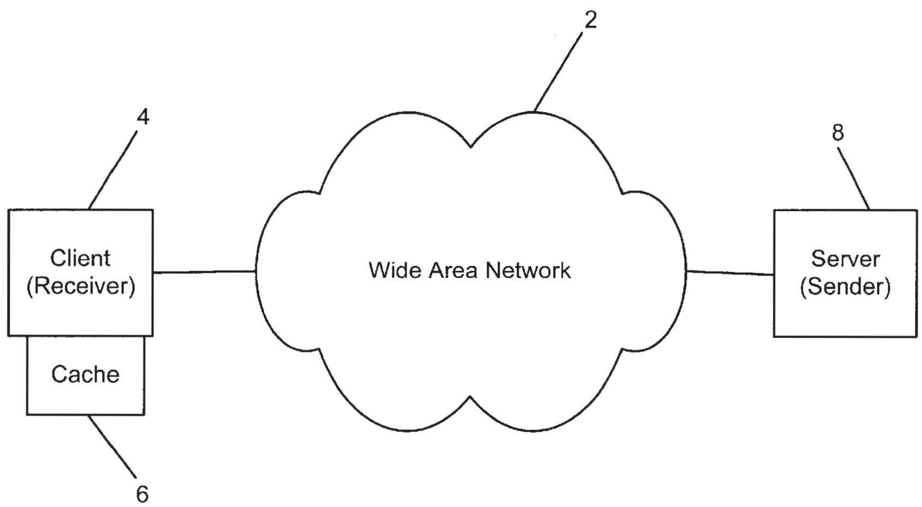


FIG. 1 (PRIOR ART)

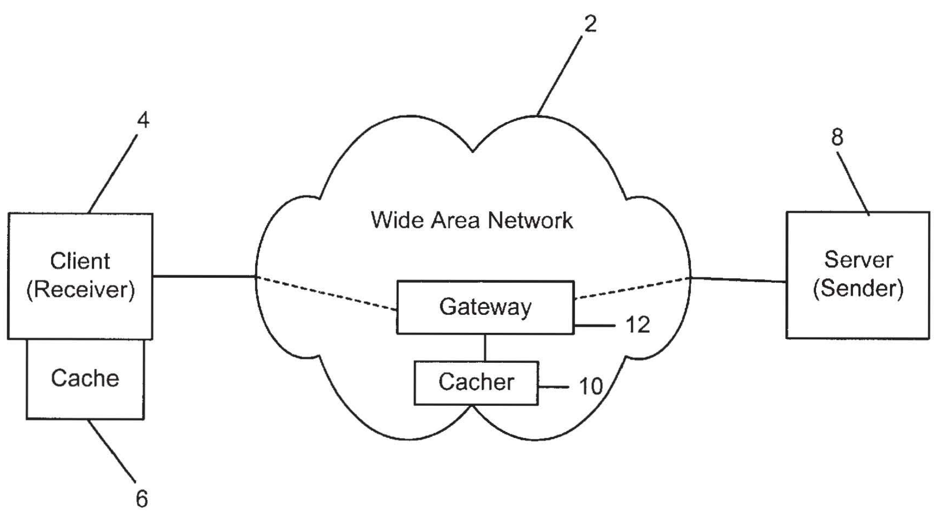


FIG. 2 (PRIOR ART)

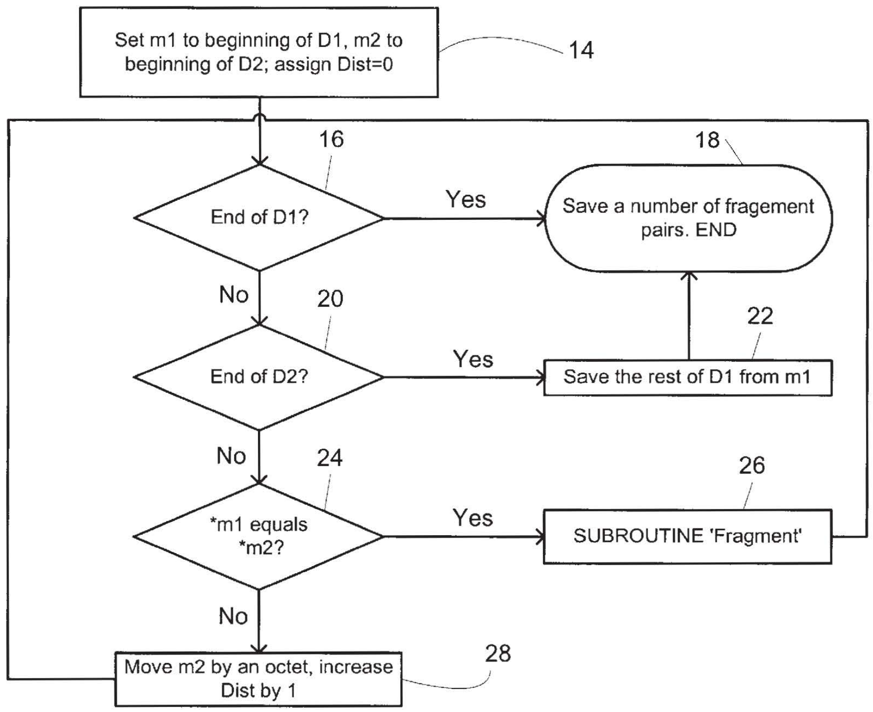
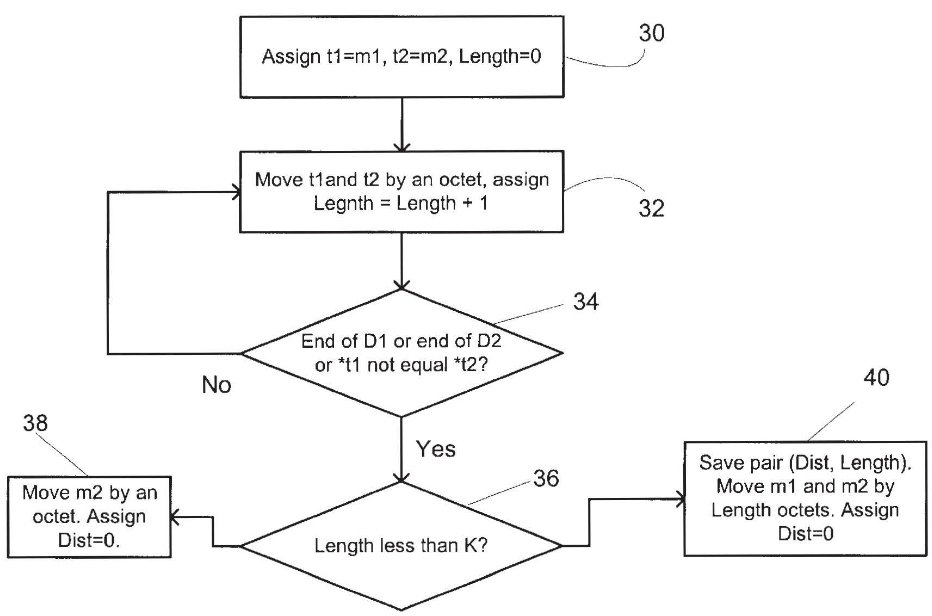


FIG. 3

SUBROUTINE 'Fragment'



Case: 14-1542 Document: 12 Page: 114 Filed: 08/08/2014

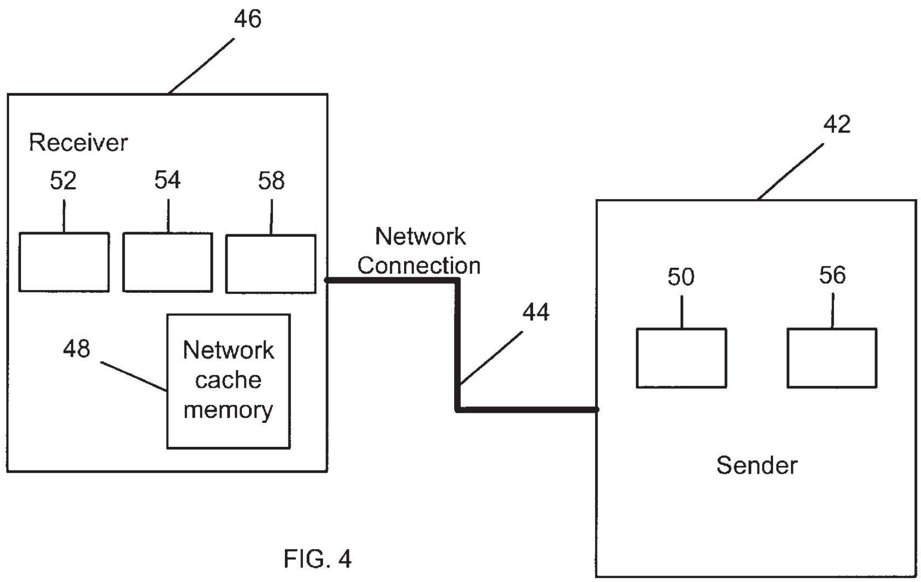


FIG. 4

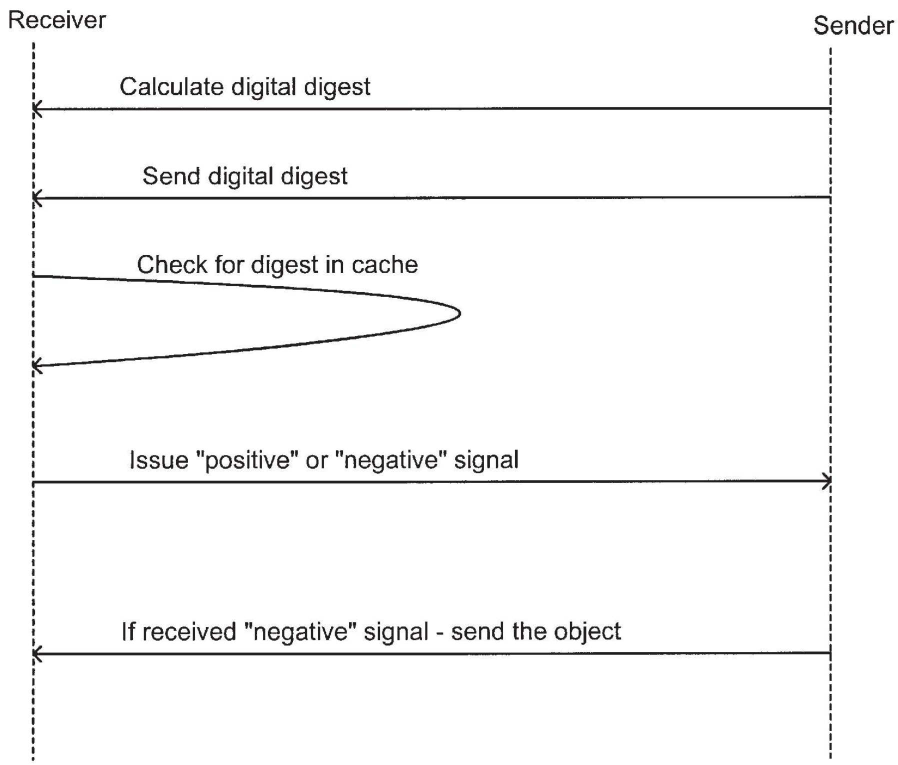


FIG. 5

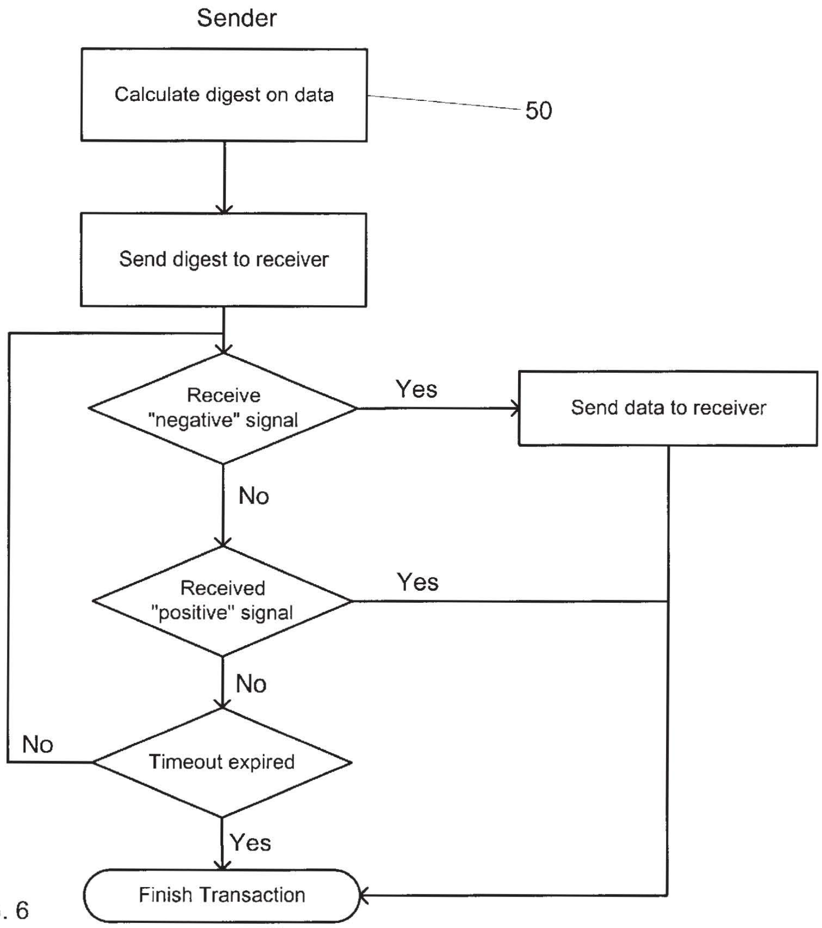

Sender
Calculate digest on data
50
Send digest to receiver
Receive "negative" signal
Yes
Send data to receiver
No
Received "positive" signal
Yes
No
No
Timeout expired
Yes
Finish Transaction

FIG. 6

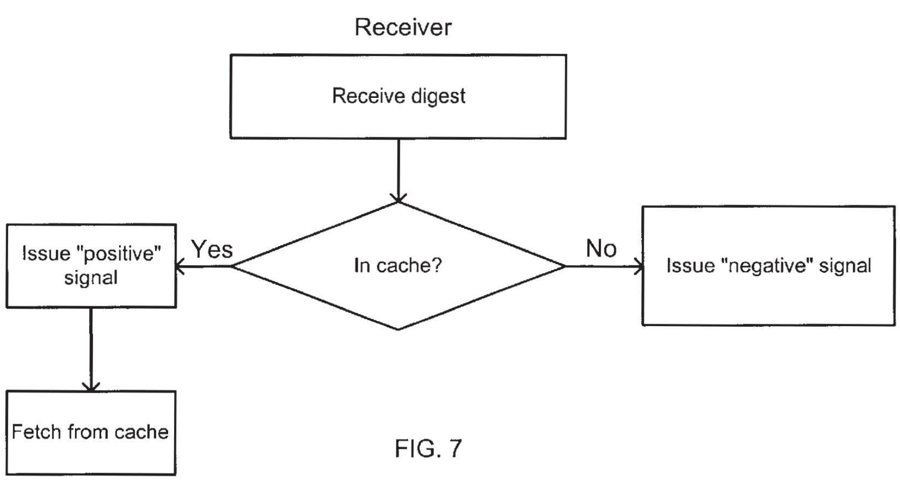

Receiver
Receive digest
In cache?
Yes
Issue "positive" signal
No
Issue "negative" signal
Fetch from cache

FIG. 7

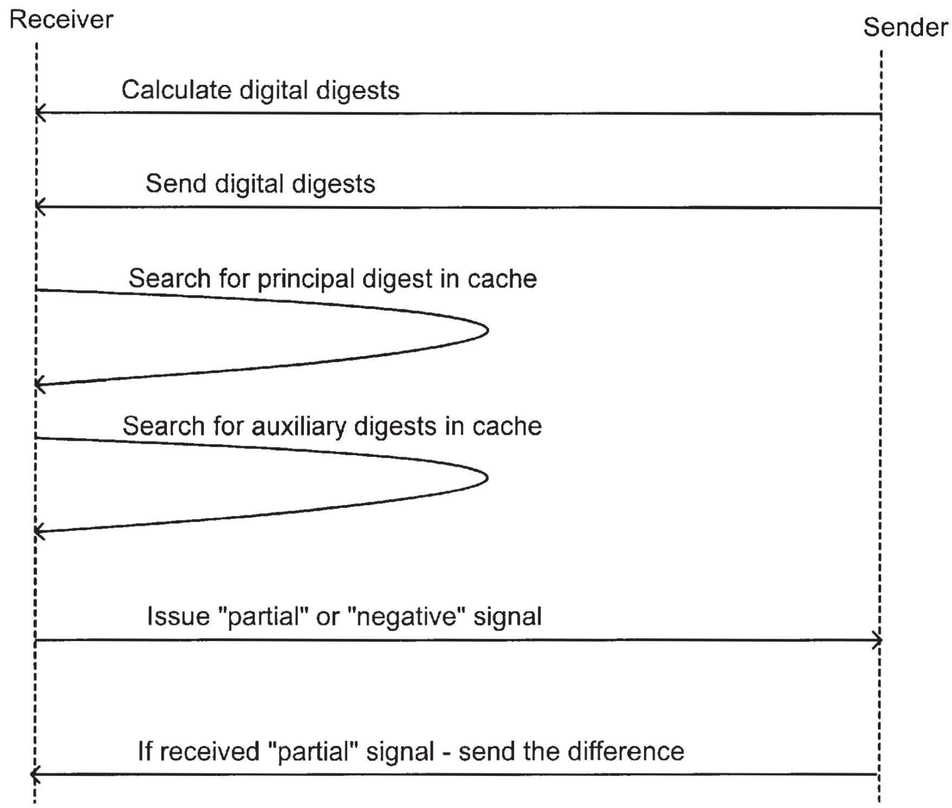

Receiver
Sender
Calculate digital digests
Send digital digests
Search for principal digest in cache
Search for auxiliary digests in cache
Issue "partial" or "negative" signal
If received "partial" signal - send the difference

FIG. 8

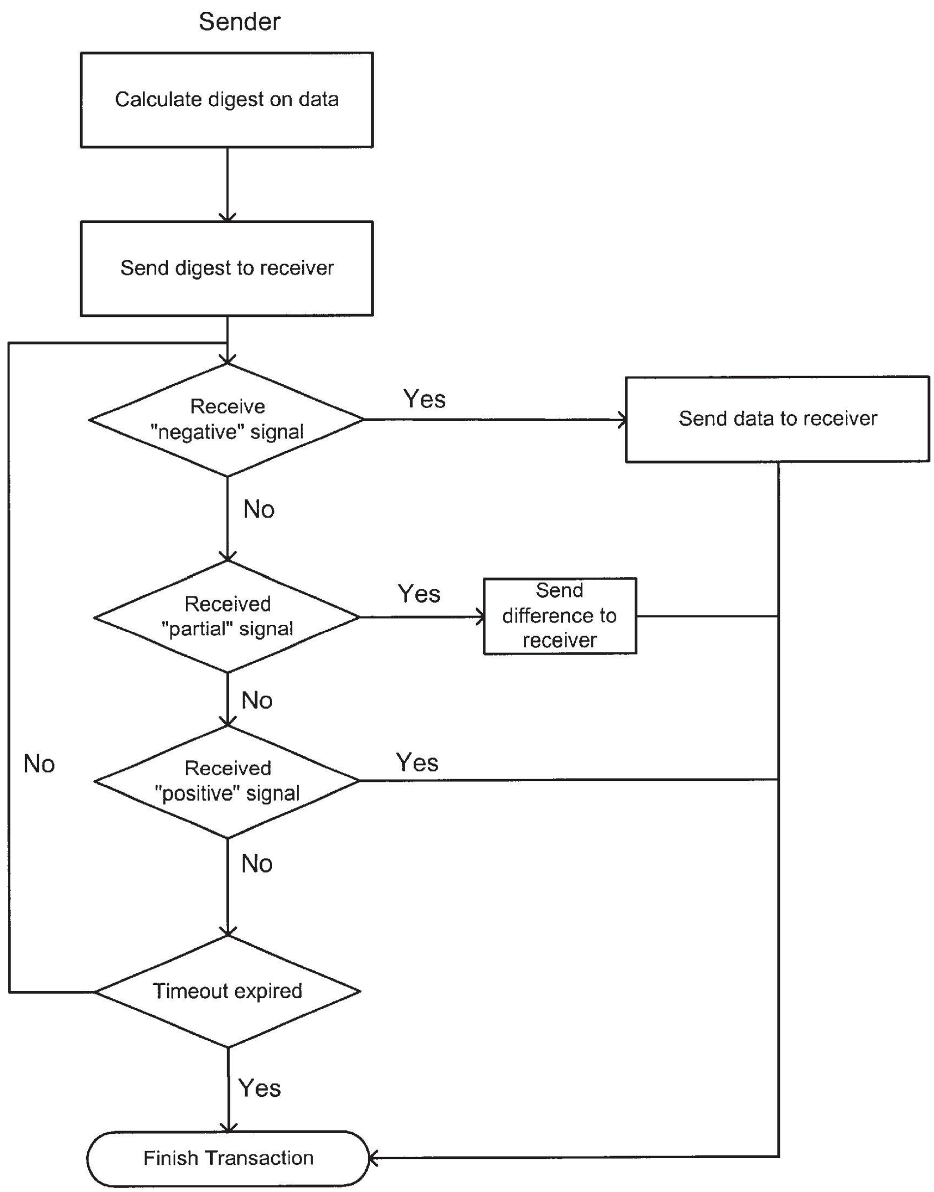

Sender
Calculate digest on data
Send digest to receiver
Receive "negative" signal
Yes
Send data to receiver
No
Received "partial" signal
Yes
Send difference to receiver
No
No
Received "positive" signal
Yes
No
Timeout expired
Yes
Finish Transaction

FIG. 9

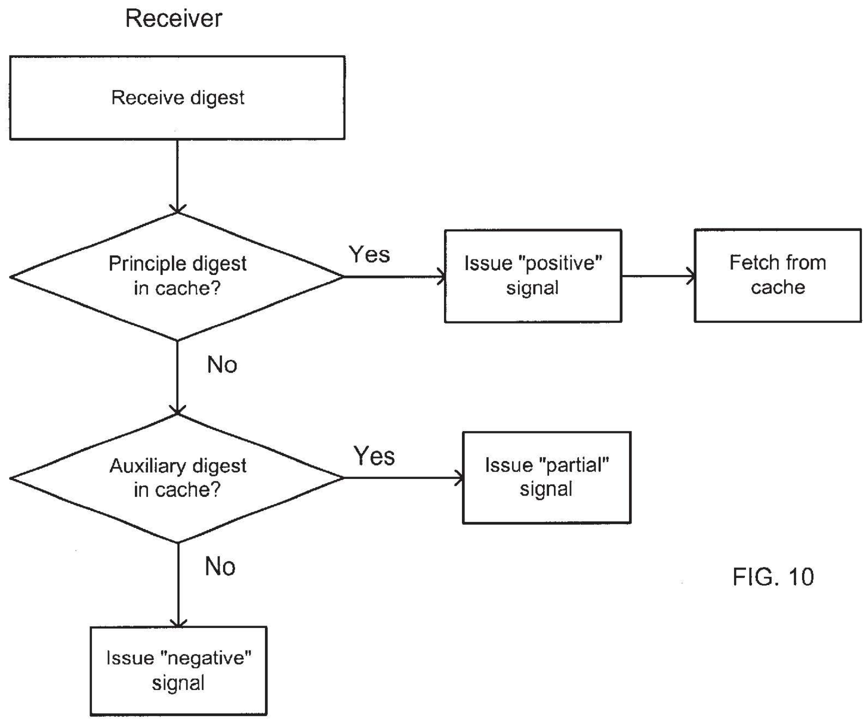

Receiver
Receive digest
Principle digest in cache?
Yes
Issue "positive" signal
Fetch from cache
No
Auxiliary digest in cache?
Yes
Issue "partial" signal
No
Issue "negative" signal

FIG. 10

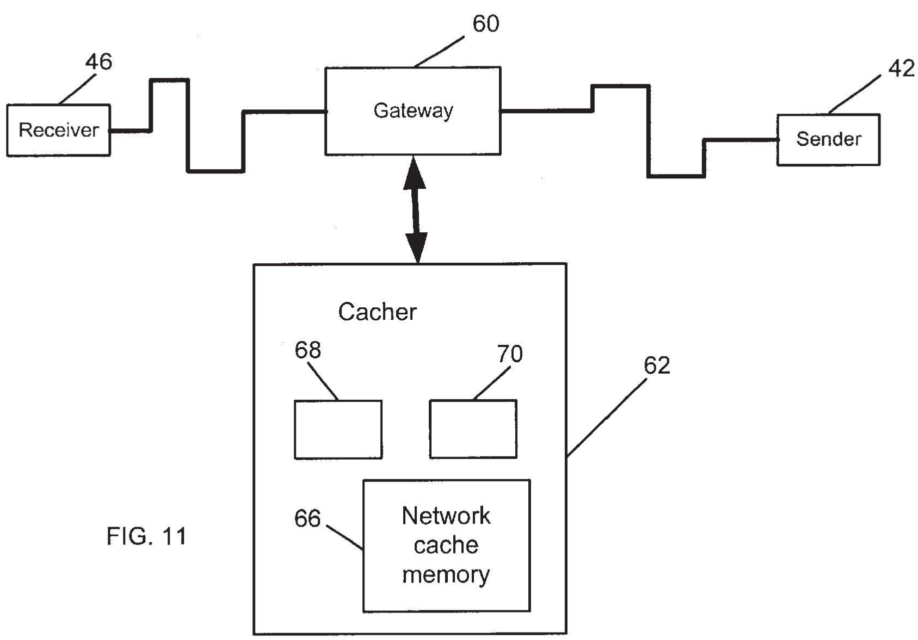

60
46
42
Receiver
Gateway
Sender
Cacher
68
70
62
66
Network cache memory

FIG. 11

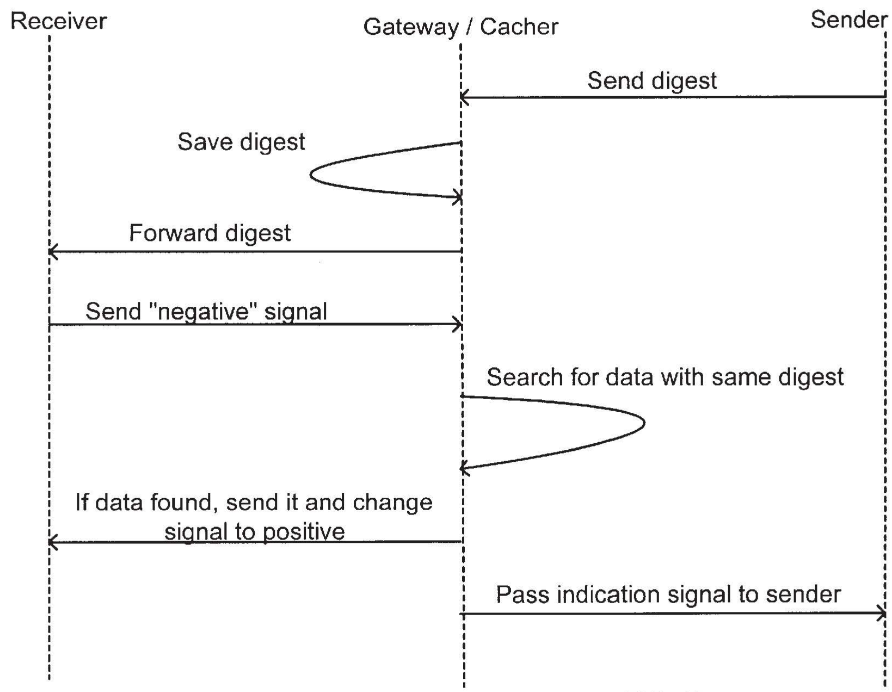

Receiver
Gateway / Cacher
Sender
Send digest
Save digest
Forward digest
Send "negative" signal
Search for data with same digest
If data found, send it and change signal to positive
Pass indication signal to sender

FIG. 12

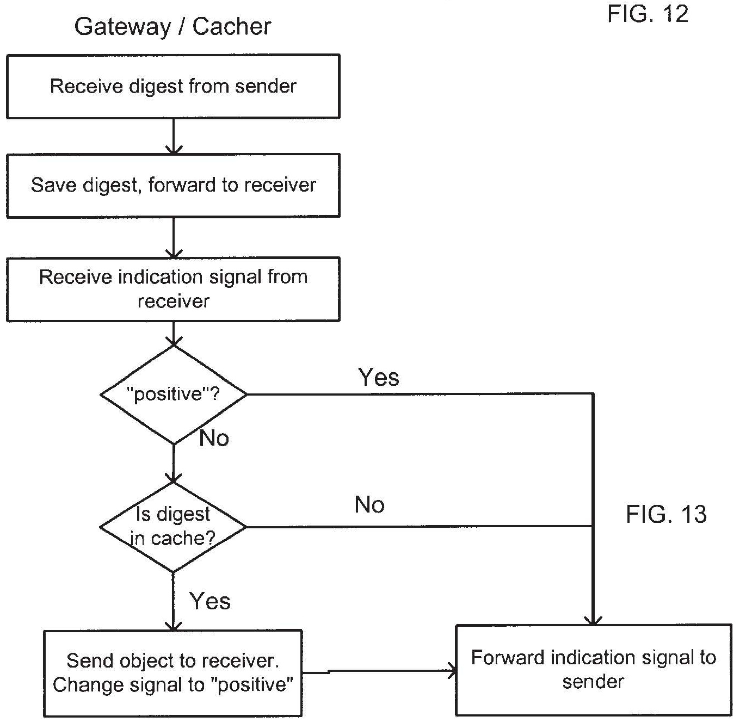

Gateway / Cacher
Receive digest from sender
Save digest, forward to receiver
Receive indication signal from receiver
"positive"?
Yes
No
Is digest in cache?
No
Yes
Send object to receiver. Change signal to "positive"
Forward indication signal to sender

FIG. 13

Case: 14-1542 Document: 12 Page: 120 Filed: 08/08/2014

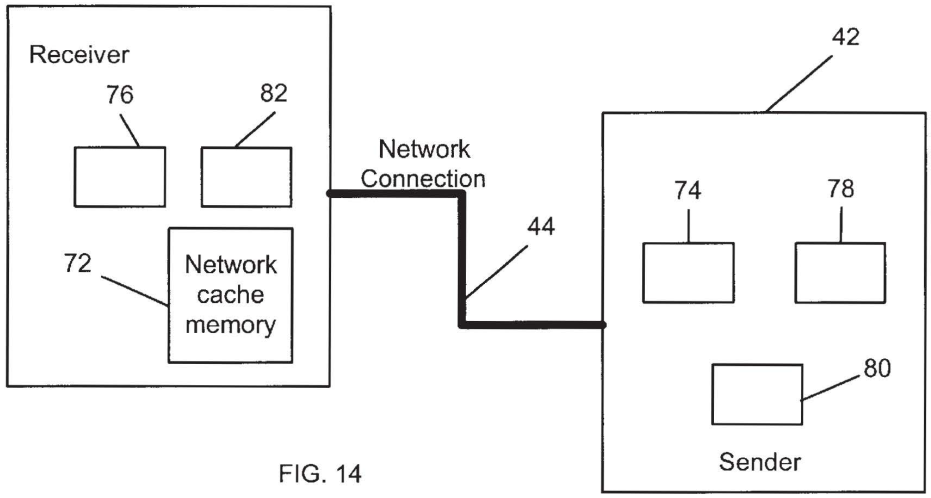


FIG. 14

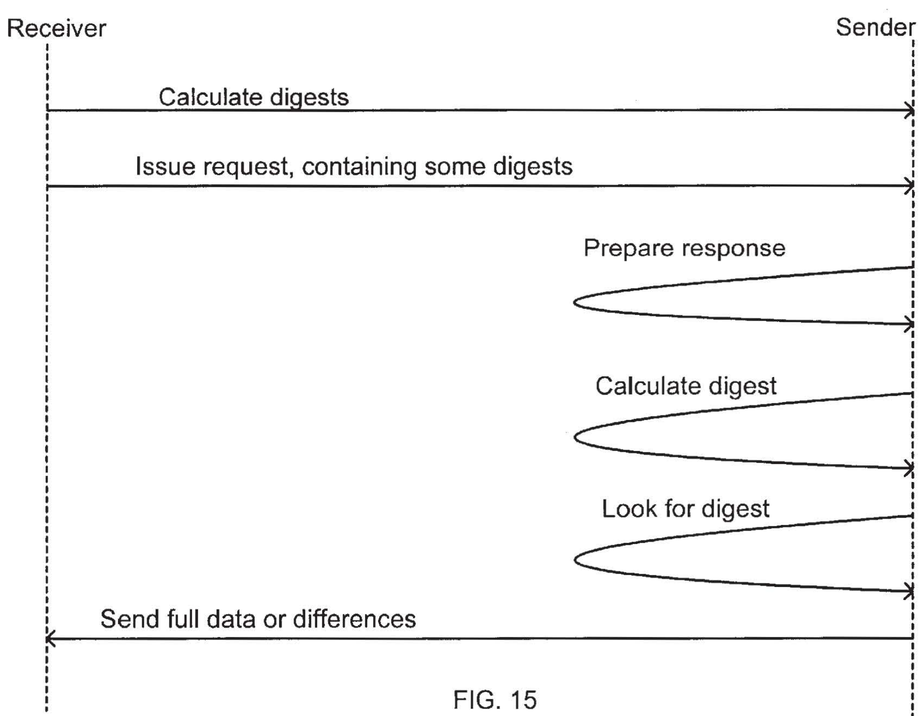


FIG. 15

SYSTEM AND METHOD FOR DATA ACCESS

RELATED APPLICATION

This application claims priority and is entitled to the filing date of Israeli application Ser. No. 126292 filed Sep. 18, 1998, and entitled "System And Method For Data Access," and which describes the same invention as defined herein.

FIELD OF THE INVENTION

The present invention relates to data access in networks. Specifically, the invention is concerned with a method, system and apparatus for increasing the speed of data accessing in communication networks.

BACKGROUND OF THE INVENTION

Many known applications and protocols provide means for caching and verifying of data transmitted via a network **2** (FIG. **1**, prior art). Thus, a client (receiver) **4** caches data received from network **2** in cache **6**. Then, when data from a remote server (sender) **8** is requested, it first searches its local cache. If the requested data is available in the cache and is verified to be valid, the client uses it, and transmission over the network is not required. Gateway or proxy caches **10** (FIG. **2**, prior art) are able to operate in a similar manner.

The most well-known techniques are as follows:

1) In response to a request from a receiver, a sender attaches to the sent data an expiration time in absolute or relative form. The receiver, and possibly proxies, cache the data together with its request until the expiration time. Then the data is retrieved from the cache. In some cases, the receiver guesses the expiration time.

The problem associated with this technique is that the data entity can be changed before the expiration time, and the receiver would use an obsolete version of the data without even knowing it. Also, when the time has expired, the data will be resent, even if it is up to date.

2) In response to a request from a receiver, the sender attaches a validator to the sent data. The validator changes at least every time the data changes; in many cases, system time is used as the validator. The receiver, and possibly proxies, cache the data together with its request. When making the next request for the same data to the same sender, the receiver includes the validator. The sender keeps track of the data and resends it only if it were changed.

The problems associated with this technique are:

a) Data is cached according to requests and senders. If the same request is directed to different servers, cached data cannot be reused.

b) Requests without concrete data cannot be cached.

c) The sender must track the cached data, which is not always possible.

None of the prior art techniques discussed above provides means for transmitting minor differences in data. Additionally, if data is retrieved through a caching proxy, there is a danger that an unauthorized user will have access to the data.

It is therefore a broad object of the present invention to provide a method, system and apparatus for increasing the speed of data access in a packet-switched network.

Another object of the present invention is to decrease data traffic throughout the network.

Still another object of the present invention is to decrease the required cache size.

A yet further object of the present invention is to maintain accessed data integrity and to improve security.

SUMMARY OF THE INVENTION

The terms "data" or "data object" as used herein refer to a file or range of octets in a file, a range of frames in a video stream or RAM-based range of octets, a transport level network packet, or the like.

The term "digital digest" as used herein refers to a fixed-size binary value calculated from arbitrary-size binary data in such a way that it depends only on the contents of the data and the low probability that two different data or objects have the same digital digest.

The term "gateway" as used herein also includes network proxies and routers.

If a sender/computer in a network is required to send data to another receiver/computer, and the receiver/computer has data with the same digital digest as that of the data to be sent, it can be assumed with sufficient probability for most practical applications that the receiver/computer has data which is exactly the same as the data to be sent. Then, the receiver/computer can use the data immediately without its actual transfer through the network. In the present invention, this idea is used in a variety of ways.

In one embodiment of the invention, a sender/computer required to send data to a receiver/computer computer initially sends a digital digest of the data. If the receiver/computer already has data with the same digital digest, it uses this data as if it were actually transmitted from the sender/computer. Additionally, digital digests for other data objects can be sent together with the principal digest. If the receiver/computer cannot find data having the principal digest, it searches for data with one of these auxiliary digests. If such data is found, the sender/computer is required to send only the difference between the requested data object and the data object corresponding to the digest.

The expression "difference between a first data or data object and a second data or data object" as used herein means any bit sequence that enables the restoration of the first data, given the second data, the bit sequence and the method employed in calculating the difference.

The invention may be implemented in a gateway system. Such a system comprises a gateway computer connected to a packet-switched network in such a way that network packets sent between at least two other computers pass through it; a caching computer connected to the gateway computer, the caching computer having a network cache memory in its permanent storage memory, means for calculating a digital digest on the data it stores and means for comparison between a digital digest calculated on data in its network cache memory and a digital digest received from the packet-switched network by the gateway computer. When this system intercepts an indication signal other than a positive indication signal for a certain digital digest from a receiver/computer computer, if it has data with the same digest, it sends this data to the receiver/computer.

In another embodiment of the present invention, a client computer sends to a server computer a request including digital digests. A sender/computer forming a response then searches for data with the same digital digests as those received. If the digest of the data in the response equals one of the received digests, the server only sends confirmation. If the digest of another data is identical to one of the received digests, only the difference(s) between these data is sent.

In accordance with the present invention, there is therefore provided a system for data access in a packet-switched

network, comprising a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, said sender/computer and said receiver/computer communicating through said network; said sender/computer further including means for calculating digital digests on data; said receiver/computer further including a network cache memory and means for calculating digital digests on data in said network cache memory; and said receiver/computer and/or said sender/computer including means for comparison between digital digests.

The invention also provides a system for data access in a packet-switched network, comprising a gateway computer including an operating unit, a memory and a processor connected to said packet-switched network in such a way that network packets sent between at least two other computers pass through it; a caching computer including an operating unit, a first memory, a permanent storage memory and a processor connected to said gateway computer through a fast local network; said caching computer further including a network cache memory in its permanent storage memory, means for calculating a digital digest on data stored therein and means for comparison between a digital digest calculated on data in its network cache memory and a digital digest received from said packet-switched network through said gateway computer.

In addition, the invention provides a system for data access in a packet-switched network, comprising a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, said sender/computer and said receiver/computer communicating through a network; said sender/computer further including means for calculating digital digests on data, and said receiver/computer further including a network cache memory, means for storing a digital digest received from said network in its permanent storage memory and means for comparison between digital digests.

The invention further provides a method performed by a sender/computer in a packet-switched network for increasing data access, said sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and said sender/computer being operative to transmit data to a receiver/computer, the method comprising the steps of transmitting a digital digest of said data from said sender/computer to said receiver/computer; receiving a response signal from said receiver/computer at said sender/computer, said response signal containing a positive, partial or negative indication signal for said digital digest, and if a negative indication signal is received, transmitting said data from said sender/computer to said receiver/computer.

The invention still further provides a method for increasing data access performed by a sender/computer in a packet-switched network, said sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and said sender/computer being operative to transmit principal data to a receiver/computer, said method comprising the steps of transmitting digital digests of said principal data and of one or more auxiliary data from said sender/computer to said receiver/computer; receiving a response signal at said sender/computer from said receiver/computer, said response signal containing a positive, negative or partial indication signal, and if a partial indication signal is received, said sender/computer transmitting a signal constituting the difference between said principal data and corresponding auxiliary data.

The invention yet further provides a method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of receiving a message containing a digital digest from said network; searching for data with the same digital digest in said network cache memory, and if data having the same digital digest as the digital digest received is not uncovered, forming a negative indication signal and transmitting it back through said network.

Still further, the invention provides a method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of receiving a message containing a digital digest from said network; searching for data with the same digital digest in said network cache memory, and if data having the same digital digest as the digital digest received is uncovered, forming a positive indication signal and transmitting it back through said network.

In addition, the invention provides a method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of receiving a message containing a principal digital digest and one or more auxiliary digital digests from said network; searching in predetermined locations in said permanent storage memory for data with a digital digest substantially identical to said principal digital digest; searching in predetermined locations in said permanent storage memory for data with a digital digest substantially identical to one of said auxiliary digital digests; and if data having the same digital digest as one of said auxiliary digital digests received is uncovered, forming a partial indication signal and transmitting it back through said network.

Yet further, the invention provides a method for increased data access performed by a computer system in a packet-switched network, said computer system including a network cache memory and being operationally interposed between a sender/computer and a receiver/computer so that data packets sent between said sender/computer and said receiver/computer are delivered through said computer system; said method comprising the steps of intercepting a message containing a digital digest transmitted from said sender/computer to said receiver/computer, and transmitting data with a digital digest substantially identical to the digital digest received from said sender/computer to said receiver/computer.

In addition, the invention provides a method for increased data access performed by a computer system in a packet-switched network, said computer system including a network cache memory and being operationally interposed between a sender/computer and a receiver/computer so that data packets sent between said sender/computer and said receiver/computer are delivered through said computer system; said method comprising the steps of intercepting a message containing a digital digest transmitted from said sender/computer to said receiver/computer; intercepting a message containing an indication signal other than a positive indication signal transmitted from said receiver/computer to said sender/computer in response to said message containing a digital digest, and transmitting data with a digital digest substantially identical to the digital digest received from said sender/computer to said receiver/computer.

Additionally, the invention provides a method for increased data access performed by a client computer in a packet-switched network, said client computer including an operating unit, a first memory and a processor, said method comprising the steps of sending a request for data from said client computer to a server, said request containing digital digests for different data; said server preparing a response to said request, searching for data with a digital digest substantially identical to one of the digital digests received in said request, and producing the difference between said response and the uncovered data.

Finally, the invention provides apparatus for increased data access in a packet-switched network, comprising a computer connected to said packet-switched network, including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory; means for calculating digital digests of data in said network cache memory; means for comparison between digital digests, and means for sending the results of comparison between a digital digest received from another computer in said network and a digital digest calculated on data in said network cache memory back to said other computer.

The invention will now be described in connection with certain preferred embodiments with reference to the following illustrative figures so that it may be more fully understood.

With specific reference now to the figures in detail, it is stressed that the particulars shown are by way of example and for purposes of illustrative discussion of the preferred embodiments of the present invention only, and are presented in the cause of providing what is believed to be the most useful and readily understood description of the principles and conceptual aspects of the invention. In this regard, no attempt is made to show structural details of the invention in more detail than is necessary for a fundamental understanding of the invention, the description taken with the drawings making apparent to those skilled in the art how the several forms of the invention may be embodied in practice.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a prior art wide-area network;

FIG. **2** illustrates a prior art wide-area network with a caching gateway;

FIG. **3** is a flow diagram of the method of calculating the difference between two data digests according to the present invention;

FIG. **4** is a block diagram of a first embodiment of a sender/computer-receiver/computer system according to the present invention;

FIG. **5** is a schematic representation illustrating the interaction between a sender/computer and a receiver/computer according to the system of FIG. **4**;

FIG. **6** is a flow diagram illustrating the method of operating the sender/computer according to the present invention;

FIG. **7** is a flow diagram illustrating the method of operating the receiver/computer according to the present invention;

FIG. **8** is a schematic representation illustrating the interaction between a sender/computer and a receiver/computer according to another embodiment of the present invention;

FIG. **9** is a flow diagram illustrating the method of operating the sender/computer according to a further embodiment of the present invention;

FIG. **10** is a flow diagram illustrating the method of operating the receiver/computer according to the embodiment of FIG. **9**;

FIG. **11** is a block diagram of the configuration of the gateway system according to the present invention;

FIG. **12** is a schematic representation of the interaction between a sender/computer, a receiver/computer, and the gateway configuration according to the present invention;

FIG. **13** is a flow diagram of the operation of the gateway;

FIG. **14** is a block diagram of a further configuration of a sender/computer-receiver/computer system according to the present invention; and

FIG. **15** is a schematic representation of the interaction between the sender/computer-receiver/computer system of FIG. **14**.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The performance gains realized by the present invention are derived from the fact that computers in common wide-area networks tend to repetitively transmit the same data over the network.

The operations described herein may take the form of electrical or optical signals. The packet-switched network may be Internet.

The term "digital digest" as used herein refers to the per se known MD5 algorithm, described in RFC 1321 by R. Rivest, which is a preferred calculation method. Other algorithms may, however, just as well be used. For example, a digital digest may be calculated according to the CRC algorithm, or by applying the CRC algorithm to different subsets or different reorderings of data, or by consecutively applying CRC and MD5. In addition, any other algorithm may be used, provided that it produces a fixed-size binary value calculated from arbitrarily-sized binary data in such a way that it depends only on the contents of said data and that the probability of two different data having the same digital digest, is low.

Whenever means for calculating the difference between two data are mentioned herein, the method as shown in FIG. **3** may be employed. The data are referred to as D**1** and D**2**. The difference between them consists of three parts: the number of fragment pairs, the array of fragment pairs, and the remainder of D**1**. A fragment pair is a pair representing the distance from the beginning of this fragment to the end of the previous one, and the number of octets in the fragment (Dist,Length). A marker m**1** is set at the beginning of the data D**1** and a marker m**2** at the beginning of D**2**.

An octet m**1** is designated as *m**1** and an octet m**2** as *m**2**. An integer K>1, which represents a minimal length of a fragment encoded, e.g., K=3, is chosen.

As stated above, m**1** is set at the beginning of D**1**, m**2** at the beginning of D**2**, and Dist=0 is assigned at 14. A loop is then entered: if m**1** is at the end of D**1** (**16**), a number of fragment pairs is saved at 18, and the algorithm is completed. If m**2** is at the end of D**2** (**20**), the rest of D**1** from m**1** is saved at 22, a number of fragment pairs is saved, and the algorithm is completed. If *m**1** equals *m**2** (**24**), a subroutine "Fragment" is entered at 26; otherwise, m**2** is moved by one octet toward the end of D**2** and Dist is increased by 1 at 28.

The subroutine "Fragment" proceeds as follows: New markers t1=m1 and t**2**=m**2** are set and Length=0 is assigned at 30. t**1** and t**2** are moved by an octet toward the ends of D**1** and D**2** and Length is increased at 32. If t**1** is at the end of D**1**, or t**2** is at the end of D**2**, or *t**1** does not equal *t**2** at the end of the fragment (**34**), then the Length is a length of the fragment and Dist is the distance between the beginning of

this fragment and the end of the previous one. If the Length <K as determined by 36, the fragment is dropped at 38, m**2** is moved by one octet and the subroutine is terminated. Otherwise, the pair (Dist,Length) is saved, the number of pairs is increased by one, m**1** and m**2** are moved by Length octets toward the ends of D**1** and D**2**, and Dist is reset to 0 at 40. The subroutine is ended.

The sequence of fragment pairs may be further reduced in size by using the per se known Huffman encoding or by using an arithmetic coding, e.g., as disclosed in U.S. Pat. No. 4,122,440.

Restoration of the data is simple. Marker m**2** is set at the beginning of the known D**2**. Then for each fragment pair (Dist, Length) from the known difference, m**2** is moved by Dist octets, Length octets are copied from m**2** to D**1** and m**2** is moved by Length. Then the rest of D**1** is copied from the remainder

An embodiment of a sender/computer-receiver/computer system according to the present invention is schematically illustrated in FIG. **4**. A preferred embodiment is a network computer system having at least two computers. A sender/computer **42** (also referred to herein as "sender/computer") having an operating unit, a first memory, a permanent storage memory and a processor, is connected to the network by any network connection **44**. A remote receiver/computer **46** (also referred to herein as "receiver/computer") having an operating unit, a first memory, a permanent storage memory and a processor, is also connected to the network. The receiver/computer **46** uses a part of its permanent storage memory or its first memory, or both, as network cache memory **48**. The sender/computer has calculation means **50** for calculating a digital digest on data in its first memory or in its permanent storage memory. Similarly, the receiver/computer has calculating means **52** for calculating a digital digest on data stored in its network cache memory **48**. The receiver/computer also has comparison means **54** for comparing between such a calculated digital digest and a digital digest received from the network.

An example of a first memory could be a RAM; an example of a permanent storage memory may be a disk drive, a flash RAM or a bubble memory.

It is possible to modify this system in different ways. The receiver/computer **46** and sender/computer **42** may each include means for storing the calculated digital digest in its first memory or permanent storage memory. Additionally, the receiver/computer **46** may have means for calculating a digital digest on data in its permanent storage memory outside of its cache memory. Furthermore, the system may be modified in such a way that the sender/computer **42** has means **56** for calculating the difference between two data objects.

Interaction between the receiver/computer and the sender/computer is depicted in FIGS. **5** to **7**. The data sender/computer **42** calculates a digital digest on the data in means **50** and then transmits the calculated digest to receiver/computer **46**. The receiver/computer receives the digital digest from sender/computer **42** and then searches its network cache memory **48** for data with the same digest. If it finds such data, it uses it as if it were received from the sender/computer **42** and issues a positive indication signal to the sender/computer. Otherwise, it sends a negative indication signal to the sender/computer. Upon receiving a negative indication signal, the sender/computer transmits the data. Upon receiving a positive indication signal, or upon expiration of a predefined period of time, the sender/computer completes the transaction. This transaction begins with a receiver/computer sending a request to the sender/computer.

The above-described method may be modified in different ways. For example, absence of a signal from the receiver/computer for a predetermined period of time may be considered by the sender/computer to be a negative indication signal. Alternatively, the digital digests for some data may be stored in the permanent storage memory of the sender/computer and obtained from there, or a plurality of data may be processed in one transaction, a digital digest being calculated for each data object and a separate indication signal issued on each digital digest.

Another method of interaction between the receiver/computer **46** and the sender/computer **42** is illustrated in FIGS. **8**–**10**. The data sender/computer calculates a digital digest on the data to be sent (hereinafter, "principal digest") and for one or more other data objects (hereinafter, "auxiliary digests"). Without limiting the scope of the invention, the following data objects may be recommended: (a) a previous version of the data requested; (b) a file similar to the data requested. Then the sender/computer sends the principal and auxiliary digests to the receiver/computer. Upon receiving a message with these digital digests from the sender/computer, the receiver/computer searches its network cache memory **48** for data having the principal digest. If such data is found, it uses it as if it were received from sender/computer **42** and issues a positive indication signal to the sender/computer. Otherwise, receiver/computer **46** searches its network cache memory **48** for data with the auxiliary digests. If it finds data with a digital digest substantially equal to one of the auxiliary digests, it issues a partial indication signal to the sender/computer, with a reference to the digest. Otherwise, it issues a negative indication signal to the sender/computer. Upon receiving a negative indication signal, the sender/computer sends the data. Upon receiving a partial indication signal, the sender/computer transmits the difference between the digital digest of the data required to be sent and that of the data whose digital digest was found by the receiver/computer. This transaction may also begin with the receiver/computer sending a request to the sender/computer.

A modification of the above method is possible. For example, absence of the indication signal from the receiver/computer for a predefined period of time may be considered by the sender/computer as a negative indication signal, or the digital digests for some data may be stored in the permanent storage memory of the sender/computer and obtained from there instead of being calculated immediately before the transaction. Alternatively, a plurality of data may be processed in one transaction; a digital digest is calculated for each data object and a separate indication signal issued on every digital digest. Still alternatively, receiver/computer **46** may search not only in its network cache memory **48**, but also in predefined locations in its permanent storage memory. Sender/computer **42** may add to a digest it sends to the receiver/computer information about the possible location of the data with that digital digest in the receiver/computer's permanent storage memory.

Another embodiment of the present invention is schematically illustrated in FIG. **11**. Shown is a system comprising a gateway computer or gateway **60** including an operating unit, a first memory and a processor, and a caching computer **62** including an operating unit, a first memory, a permanent storage memory and a processor, connected to the gateway **60** through any fast network connection **64**, e.g., Ethernet. Gateway **60** is connected to a wide-area packet-switched network in such a way that network packets sent between at least two other computers **42** and **46** pass through the gateway **60**. The caching computer **62** uses a part of its

permanent storage memory for network cache memory **66**. Caching computer **62** has means **68** for calculating the digital digest of data in its network cache memory **66**, and means **70** for comparison between such a calculated digital digest and a digital digest received by gateway computer **60** from the wide-area network. It should be noted that gateway computer **60** may be integrally formed with the caching computer. The caching computer may have means for storing a calculated digital digest in its first memory or permanent storage memory.

By way of example, operations which may be performed in such a system will now be described with reference to FIGS. **12** and **13**. The gateway **60**, operationally interposed between a sender/computer **42** and a receiver/computer **46**, intercepts a digital digest sent from the sender/computer to the receiver/computer, saves it in its memory, and passes it unchanged to the receiver/computer **46**. Then the gateway **60** intercepts an indication signal other than a positive indication signal issued by the receiver/computer. If there was a digest for this indication signal, the caching computer **62** searches for data with the same digital digest in its network cache memory **66**. If that digest is found, then the gateway sends the data to the receiver/computer, changes the indication signal to positive, and then passes the indication signal to sender/computer **42**.

Further, the caching computer **62** may verify a digital digest for a data object stored in its network cache memory **66** by calculating the digital digest for that data and comparing it to the digest stored in the network cache memory. The calculated digital digest may be stored in the network cache memory **66** and the data object-digital digest pair may be marked as not requiring further verification.

Another further embodiment of the present invention is schematically illustrated in FIG. **14**. It consists of a network computer system comprising at least two computers: a sender/computer **42** including an operating unit, a first memory, a permanent storage memory and a processor which is connected to a network **44**. A remote receiver/computer **46** having an operating unit, a first memory, a permanent storage memory and a processor is also connected to the network. The receiver/computer uses a part of its permanent storage memory or its first memory, or both, for network cache memory **72**. The sender/computer **42** has means **74** for calculating a digital digest for data in its memory or in its permanent storage. The receiver/computer **46** has means **76** for calculating a digital digest for data stored in its network cache memory. The sender/computer **42** also has means **78** for comparison between such a calculated digital digest and a digital digest received from the network. The sender/computer further includes means **80** for calculating the difference between two data objects, and receiver/computer **46** includes means **82** for restoring a data object from another data object and the difference between said data object being restored and said another data object.

An interaction between the sender/computer and receiver/computer according to this system is illustrated in FIG. **15**. When receiver/computer **46** is required to request data from the server or sender/computer **42**, it calculates one or more digital digests for different data objects stored in its network cache memory **72** or in its permanent storage memory. Without limiting the scope of the invention, the following data objects may be recommended: (a) a previous version of the data requested; (b) a file similar to the data requested; (c) a data set similar to the data requested, which may be generated in a first memory; (d) a large data file or database including fragments of octets, similar to the data requested.

The receiver/computer then transmits a request for data, containing one or more of the above-mentioned digital digests. The sender/computer prepares a response to the request, and then calculates a digital digest on the data in the response. If the calculated digest is equal to one of the digital digests in the request, the sender/computer sends a confirmation. Otherwise, the sender/computer may continue searching for the data objects with the same digital digests in the predefined subset of its permanent storage memory. If it finds such data, it calculates the difference between this data and the data in the response, and sends only the difference. Otherwise, the sender/computer sends the response as prepared.

Variations of the above method are envisioned. For example, a number of requests for data may be sent simultaneously. The digital digests on the receiver/computer may be calculated earlier and stored in the permanent memory of the receiver/computer. The digital digests on the sender/computer may also be calculated earlier and stored in the permanent memory of the sender/computer.

It will be evident to those skilled in the art that the invention is not limited to the details of the foregoing illustrated embodiments and that the present invention may be embodied in other specific forms without departing from the spirit or essential attributes thereof. The present embodiments are therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description, and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

What is claimed is:

1. A system for data access in a packet-switched network, comprising:

a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, said sender/computer and said receiver/computer communicating through said network;

said sender/computer further including means for creating digital digests on data;

said receiver/computer further including a network cache memory and means for creating digital digests on data in said network cache memory; and

said receiver/computer including means for comparison between digital digests.

2. The system as claimed in claim **1**, wherein said receiver/computer further includes means for a digital digest for data stored in said permanent storage memory.

3. The system as claimed in claim **1**, wherein said receiver/computer further includes means for storing said created digital digest in its first or permanent memory.

4. The system as claimed in claim **1**, wherein said sender/computer further includes means for the difference between two data objects and said receiver/computer further includes means for restoring a data object from another data object and the difference between said data object being restored and said another data object.

5. The system as claimed in claim **1**, wherein said sender/computer further includes means for the difference between two data objects and said receiver/computer further includes means for restoring a data object from another data object and the difference between said data object being restored and said another data object.

6. A system for data access in a packet-switched network, comprising:

a gateway including an operating unit, a memory and a processor connected to said packet-switched network

in such a way that network packets sent between at least two other computers pass through it;

a caching computer connected to said gateway through a fast local network, wherein said caching computer includes an operating unit, a first memory, a permanent storage memory and a processor;

said caching computer further including a network cache memory in its permanent storage memory, means for a digital digest and means for comparison between a digital digest on data in its network cache memory and a digital digest received from said packet-switched network through said gateway.

7. The system as claimed in claim **6**, wherein said caching computer further includes means for a digital digest for data in its network cache memory.

8. The system as claimed in claim **6**, wherein said caching computer is integrally formed with said gateway.

9. The system as claimed in claim **6**, wherein said caching computer further includes means for storing said digital digest in said permanent storage memory.

10. A system for data access in a packet-switched network, comprising:

a sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and a remote receiver/computer including an operating unit, a first memory, a permanent storage memory and a processor, said sender/computer and said receiver/computer communicating through a network;

said sender/computer further including means for creating digital digests on data, and

said receiver/computer further including a network cache memory, means for storing a digital digest received from said network in its permanent storage memory and means for comparison between digital digests.

11. A method performed by a sender/computer in a packet-switched network for increasing data access, said sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and said sender/computer being operative to transmit data to a receiver/computer, the method comprising the steps of:

creating and transmitting a digital digest of said data from said sender/computer to said receiver/computer;

receiving a response signal from said receiver/computer at said sender/computer, said response signal containing a positive, partial or negative indication signal for said digital digest, and

if a negative indication signal is received, transmitting said data from said sender/computer to said receiver/computer.

12. The method as claimed in claim **11**, wherein said sender/computer creates said digital digest for the data before transmitting it to said receiver/computer.

13. The method as claimed in claim **12**, wherein said sender/computer transmits the data to said receiver/computer after a preset period of time has passed since transmitting said digital digest to said receiver/computer and a response signal has not been received.

14. The method as claimed in claim **12**, wherein, when a plurality of data objects is to be sent, a digital digest is sent for each of said data objects and a response signal is sent containing a separate indication signal for each of said data objects.

15. The method as claimed in claim **12**, wherein said digital digest creation comprises the step of Cyclic Redundancy Check against the contents of the data.

16. The method as claimed in claim **12**, wherein said digital digest creation comprises the step of MD**5** against the contents of the data.

17. A method for increasing data access performed by a sender/computer in a packet-switched network, said sender/computer including an operating unit, a first memory, a permanent storage memory and a processor and said sender/computer being operative to transmit principal data to a receiver/computer, said method comprising the steps of:

creating and transmitting digital digests of said principal data and of one or more auxiliary data from said sender/computer to said receiver/computer;

receiving a response signal at said sender/computer from said receiver/computer, said response signal containing a positive, negative or partial indication signal, and

if a partial indication signal is received, said sender/computer transmitting a signal constituting the difference between said principal data and corresponding auxiliary data.

18. The method as claimed in claim **17**, wherein additional information about the probable location of auxiliary data in said permanent storage memory of the receiver/computer is encoded and transmitted together with the corresponding digital digest.

19. The method as claimed in claim **17**, wherein said sender/computer creates said digital digest for data before transmitting said digital digest to said receiver/computer.

20. The method as claimed in claim **17**, wherein said digital digest is obtained from the permanent storage memory of said sender/computer.

21. The method as claimed in claim **17**, wherein said digital digest creation comprises the step of Cyclic Redundancy Check against the contents of the data.

22. A method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of:

receiving a message containing a digital digest from said network;

searching for data with the same digital digest in said network cache memory,

if data having the same digital digest as the digital digest received is not uncovered, forming a negative indication signal and transmitting it back through said network; and

creating a digital digest for data received from said network cache memory.

23. The method as claimed in claim **22**, further comprising searching in predetermined locations in said permanent storage memory for data with a digital digest substantially identical to the digital digest received from said network.

24. The method as claimed in claim **22**, wherein a plurality of digital digests for different data objects is received in the same message and an indication signal is generated separately for each of said data objects.

25. A method for increased data access performed by a receiver/computer in a packet-switched network, said receiver/computer including an operating unit, a first memory, a permanent storage memory, a processor and a network cache memory, said method comprising the steps of:

receiving a message containing a principal digital digest and one or more auxiliary digital digests from said network, wherein said auxiliary digital digests are correlated to data objects similar to the data object represented by said principal digest;

searching in predetermined locations in said permanent storage memory for data with a digital digest substantially identical to said principal digital digest;

searching in predetermined locations in said permanent storage memory for data with a digital digest substantially identical to one of said auxiliary digital digests; and

if data having the same digital digest as one of said auxiliary digital digests received is uncovered, forming a partial indication signal and transmitting it back through said network.

26. The method as claimed in claim **25**, further comprising the step of searching the network cache memory for data with said principal digital digest.

27. The method as claimed in claim **26**, further comprising the step of searching in said network cache memory for data with one of said auxiliary digital digests.

28. A method for increased data access performed by a computer system in a packet-switched network, said computer system including a network cache memory and being operationally interposed between a sender/computer and receiver/computer so that data packets sent between said sender/computer and said receiver/computer are delivered through said computer system; said method comprising the steps of:

intercepting a message containing a digital digest transmitted from said sender/computer to said receiver/computer, and

transmitting data with a digital digest substantially identical to the digital digest received from said sender/computer to said receiver/computer in response to said message, whereby said sender/computer is relieved of the burden of transmitting said data.

29. The method of claim **28** further comprising the step of receiving said data into the network cache memory prior to intercepting the message.

30. A method for increased data access performed by a computer system in a packet-switched network, said computer system including a network cache memory and being operationally interposed between a sender/computer and a receiver/computer so that data packets sent between said sender/computer and said receiver/computer are delivered through said computer system; said method comprising the steps of:

intercepting a message containing a digital digest transmitted from said sender/computer to said receiver/computer;

intercepting a message containing an indication signal other than a positive indication signal transmitted from said receiver/computer to said sender/computer in response to said message containing a digital digest, and

transmitting data with a digital digest substantially identical to the digital digest received from said sender/computer to said receiver/computer, whereby said sender/computer is relieved of the burden of transmitting said data.

31. The method of claim **30** further comprising the step of receiving said data into the network cache memory prior to intercepting the message.

32. A method for increased data access performed by a client computer in a packet-switched network, said client computer including an operating unit, a first memory and a processor, said method comprising the steps of:

sending a request for a single first data object from said client computer to a server, said request containing multiple digital digests for different data objects similar to said first data object;

said server preparing a response to said request, searching for a second data object with a digital digest substantially identical to one of the digital digests received in said request, and producing the difference between said first data object and the uncovered second data object.

33. The method as claimed in claim **32**, further comprising the step of transmitting said difference to said client computer.

34. The method as claimed in claim **33**, further comprising the step of using said difference for restoring the data from said response in said client computer.

* * * * *

OPAP IAP8
JUL 0 3 2012
PATENT & TRADEMARK OFFICE

PTO/SB/44 (09-07)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.
(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

Page _____ of _____

PATENT NO. : 6757717

APPLICATION NO.: 09/398,007

ISSUE DATE : 06/29/2004

INVENTOR(S) : Goldstein Leonid

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

10:47: Insert "calculating" between "for" and "a"

10:53: Insert "calculating" between "for" and "the"

11:08: Insert "calculating" between "for" and "a"

11:14: Insert "calculating" between "for" and "a"

11:19: Insert "calculated" between "said" and "digital"

11:66: Insert "calculation" between "MD5" and "against"

MAILING ADDRESS OF SENDER (Please do not use customer number below):

Proxyconn, Inc.
3211 S. Shannon Str.
Santa Ana, CA 92704

This collection of information is required by 37 CFR 1.322, 1.323, and 1.324. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1.0 hour to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Attention Certificate of Corrections Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,757,717 B1
APPLICATION NO. : 09/398007
DATED : June 29, 2004
INVENTOR(S) : Leonid Goldstein

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Col. 10: line 47: Insert --calculating-- between "for" and "a"

Col. 10: line 53: Insert --calculating-- between "for" and "a"

Col. 11: line 08: Insert --calculating-- between "for" and "a"

Col. 11: line 14: Insert --calculating-- between "for" and "a"

Col. 11: line 19: Insert --calculated-- between "said" and "digital"

Col. 11: line 66: Insert --calculation-- between "MD5" and "against"

Signed and Sealed this
Fourteenth Day of August, 2012

David J. Kappos

David J. Kappos
Director of the United States Patent and Trademark Office

United States Court of Appeals for the Federal Circuit

CERTIFICATE OF SERVICE

I, Robyn Cocho, being duly sworn according to law and being over the age of 18, upon my oath depose and say that:

Counsel Press was retained by KLARQUIST SPARKMAN, LLP, Attorneys for Appellant to print this document. I am an employee of Counsel Press.

On **August 8, 2014**, Counsel for Appellant has authorized me to electronically file the foregoing **Opening Brief of Appellant Microsoft Corporation** with the Clerk of Court using the CM/ECF System, which will send notice of such filing to the following registered CM/ECF users:

Matthew L. Cutler
Bryan K. Wheelock
Douglas A. Robinson
Harness, Dickey & Pierce, PLC
7700 Bonhomme, Suite 400
St. Louis, MO 63105

Principal Attorney for Cross-Appellant Proxyconn, Inc.

A courtesy copy has also been mailed to the above counsel.

Upon acceptance by the Court of the e-filed document, six paper copies will filed with the Court, via Federal Express, within the time provided in the Court's rules.

/s/Robyn Cocho
Counsel Press

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS AND TYPE STYLE REQUIREMENTS

1. This brief complies with the type-volume limitation of Federal Rule of Appellate Procedure 32(a)(7)(B).

_X_ The brief contains **7,896** words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(a)(7)(B)(iii),or

____ The brief uses a monospaced typeface and contains ______ lines of text, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(a)(7)(B)(iii).

2. This brief complies with the typeface requirements of Federal Rule of Appellate Procedure 32(a)(5) and the type style requirements of Federal Rule of Appellate Procedure 32(a)(6).

_X_ The brief has been prepared in a proportionally spaced typeface using MS Word 2013 in a 14 point Times New Roman font or

____ The brief has been prepared in a monospaced typeface using MS Word 2002 in a ___ characters per inch_________ font.

August 8, 2014
Date

/s/John D. Vandenberg________
Name: John D. Vandenberg

Counsel for Appellant
Microsoft Corporation